

Compile 2013 VS Community Workflow

Aim

To show one workflow for setting up a Windows pc for compiling OpenCPN.

The pc used was a 'win-32-x86 machine', running Windows 8.1.

If you are running 'x64' you will need to download the 64 bit installers for some of the programs used here. This applies to:

- Git
- CMake
- 7Zip

The steps will follow the instructions [here](#).

'Navigate' means using 'Windows Explorer' to find a folder or file.

'cd' or 'CD' is used in a Terminal window to 'change directory' (aka 'change folder')

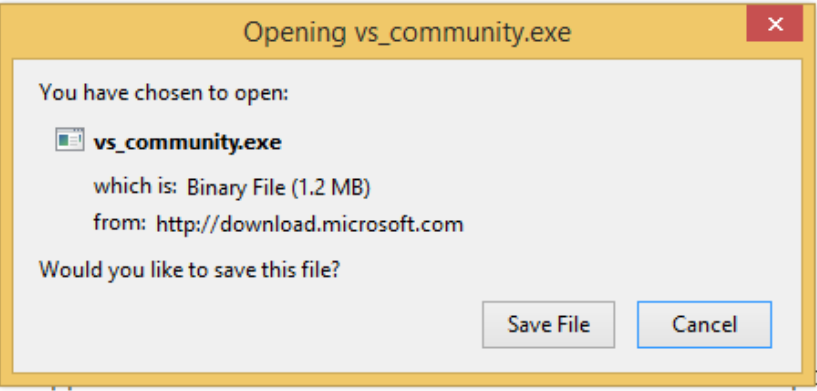
Method

- This assumes that you have accepted the suggested locations for installing the various programs.

Install Visual Studio Community 2013

<https://www.visualstudio.com/en-us/news/releases/vs2013-community-vs>

- Download the installer.



Visual Studio C

Last Update: 2/27/2017

November 12, 2014

Visual Studio Community 2013 is a new IDE to develop cross-platform solutions. Create, test, and deploy new languages, features, and development tools into this IDE. (These extensions are available from the Visual Studio Gallery.) Find out more details about Visual Studio Community 2013 [here](#).

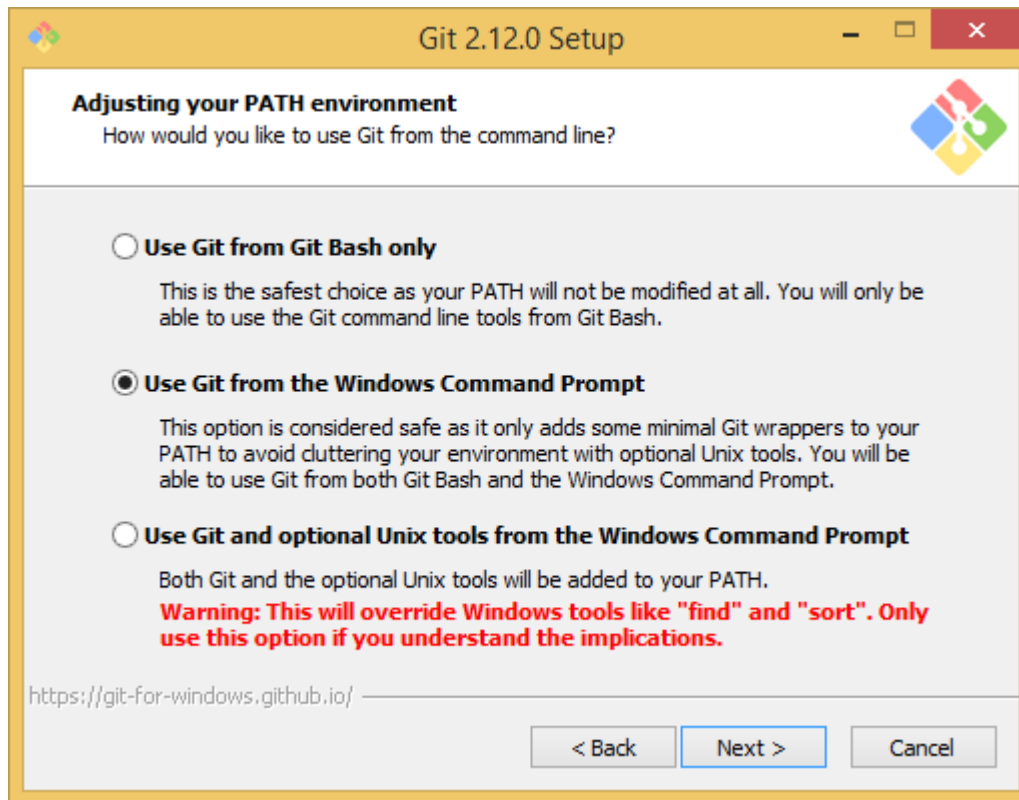
[Download Visual Studio Community 2013.](#)

- Install Visual Studio by a double-click on the '.exe'. You can disable all the optional features offered to save space.

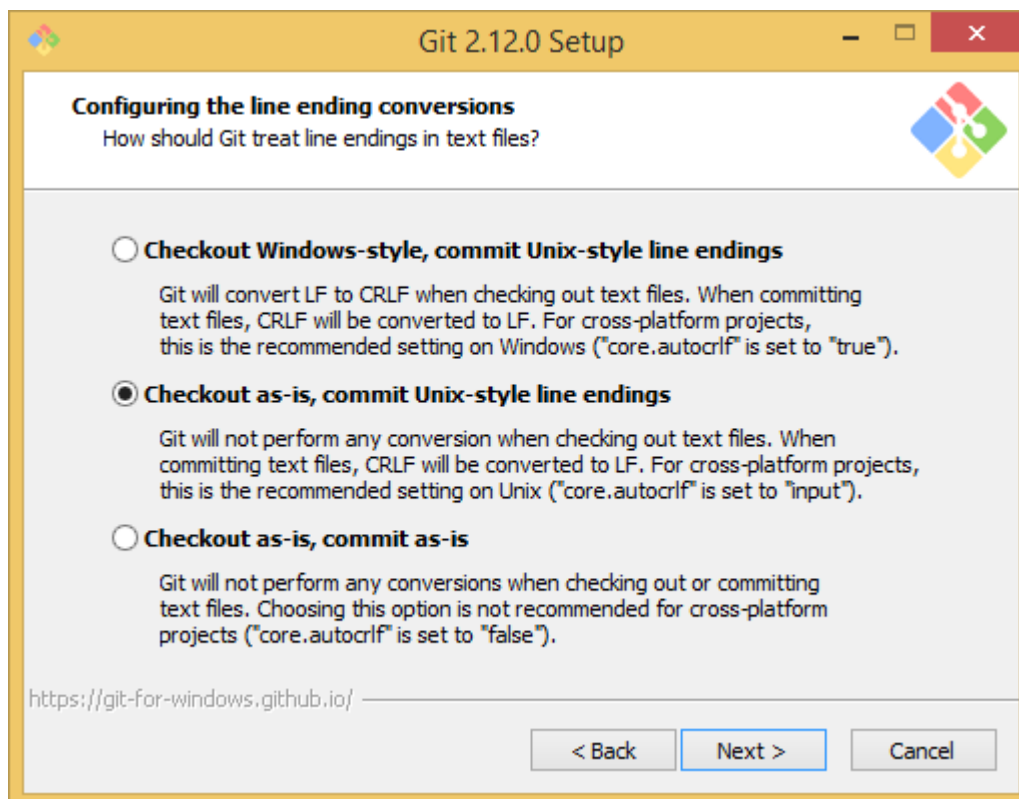
Install Git

Get the Git installation packages from <https://git-for-windows.github.io/>

- Install and let it register in the **PATH** environment variable
- The defaults for all the installation settings are fine except for just the following:
- On the **Adjusting your PATH environment**, select **Run Git from the Windows Command Prompt**.



- On the **Choosing CR/LF behavior** select **Checkout as-is, Commit Unix LF**.



- This is really important, the codebase uses Unix LF line endings and committing CR/LF makes the commits huge and hides their real content.

Install CMake


- Get the latest CMake installation packages from <http://www.cmake.org>



Download Latest Release

[Visit the download page](http://www.cmake.org)

- The installer used is shown at the bottom of this screenshot.

About ▾Resources ▾Developer Resources ▾Download

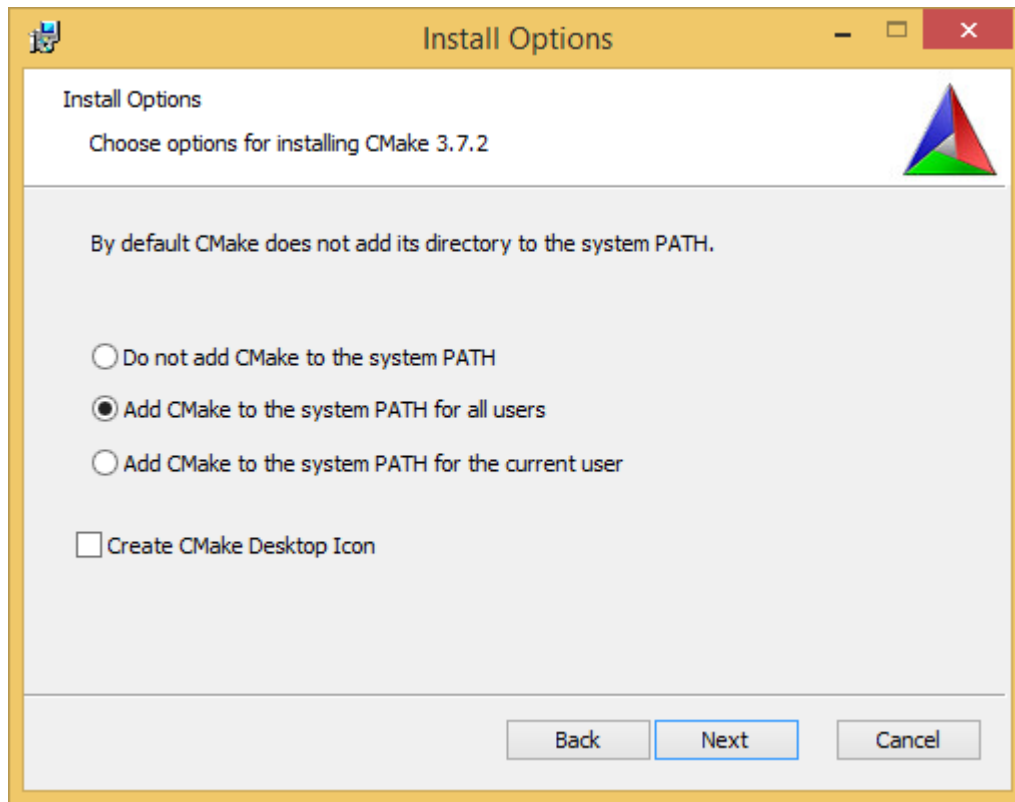
CMake 3.7 Release Notes, Source Distributions.

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.7.2.tar.gz
	cmake-3.7.2.tar.Z
Windows Source (has \r\n line feeds)	cmake-3.7.2.zip

Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.2-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.7.2-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.2-win32-x86.msi

- As part of the install choose this option:



- Install CMake and let it register the '**Path**' environment variable

Install Poedit

Get the latest 'Poedit' installation package from <http://www.poedit.net>

1. Install 'Poedit'.
2. Add '**C:\Program Files\Poedit\GettextTools\bin**' to the **PATH** environment variable.
 - Select 'System' and in the left column click 'Advanced System Settings'.

Control Panel Home

Device Manager

Remote settings

System protection

Advanced system settings

View basic information about your computer

Windows edition

Windows 8.1

© 2013 Microsoft Corporation. All rights reserved.

Get more features with a new edition of Windows



System

Processor:

Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz 3.10 GHz

Installed memory (RAM):

3.00 GB

System type:

32-bit Operating System, x64-based processor

Pen and Touch:

No Pen or Touch Input is available for this Display

Computer name, domain and workgroup settings

Computer name:

opencpn

Full computer name:

opencpn

Computer description:

Workgroup:

WORKGROUP

 [Change settings](#)

See also

Action Center

Windows Update

Windows activation

Windows is activated

[Read the Microsoft Software Licence Terms](#)

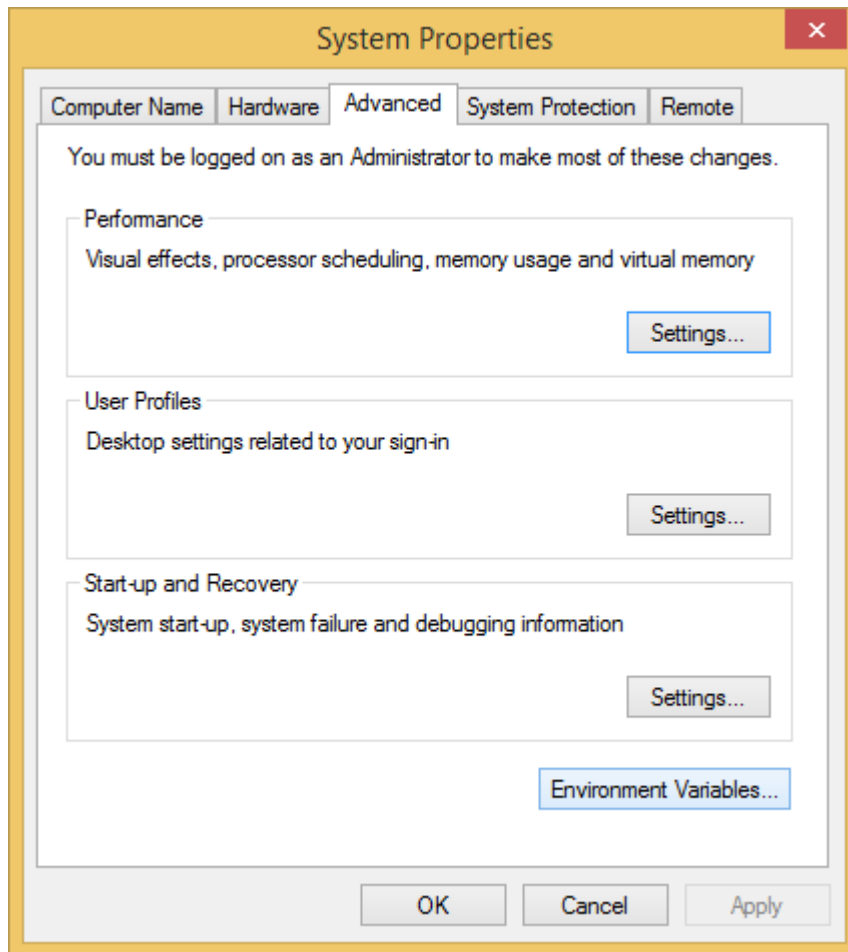
Product ID: 00259-60615-47556-AA058

[Change product key](#)

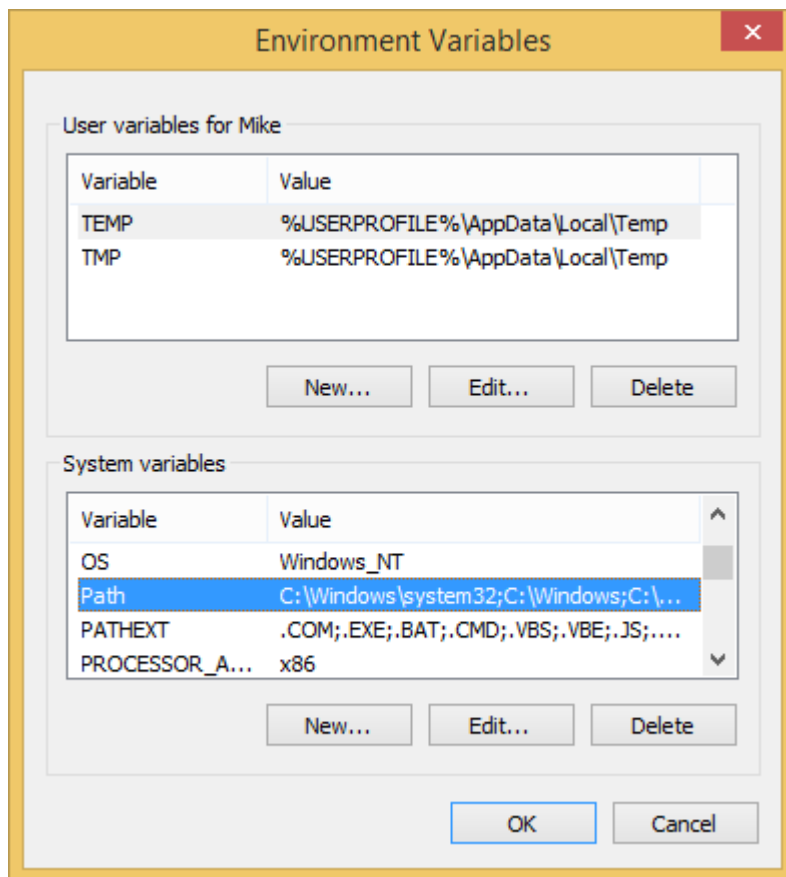
- On the Advanced tab, click on the '**Environment Variables**' button.

https://opencpn.org/wiki/dokuwiki/

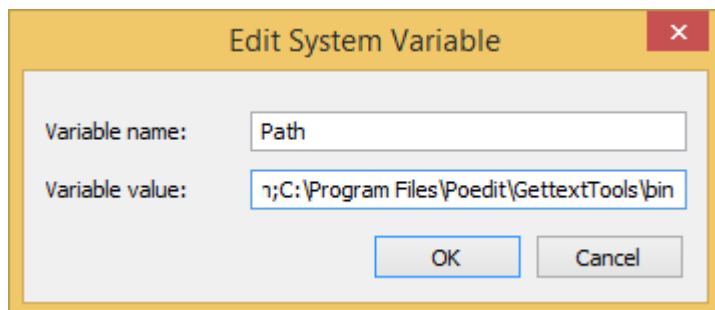
Printed on 2023/02/09 01:21



- Under '**System Variables**' find the '**Path**' system variable. Press '**Edit**'.



- In '**Variable value**' add '**;C:\Program Files\Poedit\GettextTools\bin**' to the end of the list. (The **semicolon** is important!)



- Press '**OK**' a number of times to save and exit.

Note: If you look in the '**C:\Program Files\Poedit\GettextTools\bin**' folder you will see the file '**msgfmt.exe**' which does the work of the program.

Install NSIS

In case you want to create installation packages, install NSIS Unicode 2.46.5 from <http://www.scratchpaper.com/>

- Choose the Unicode version for 'win32-x86'. All the default settings can be accepted.

**unsis**

Page 1 of 2

1 2 Next >

File	Summary + Labels	Uploaded	Size
nsis-2.46.5-Unicode-setup.exe	2.46.5 Unicode Type-Installer OpSys-Windows Featured	May 10, 2012	1.72MB

There is a “bug” in CMake, which only looks at “HKEY_LOCAL_MACHINE\SOFTWARE\NSIS” for the installation location of NSIS

The Unicode version adds its registry key in “HKEY_LOCAL_MACHINE\SOFTWARE\NSIS\Unicode”.

Some registry tweaking is needed.

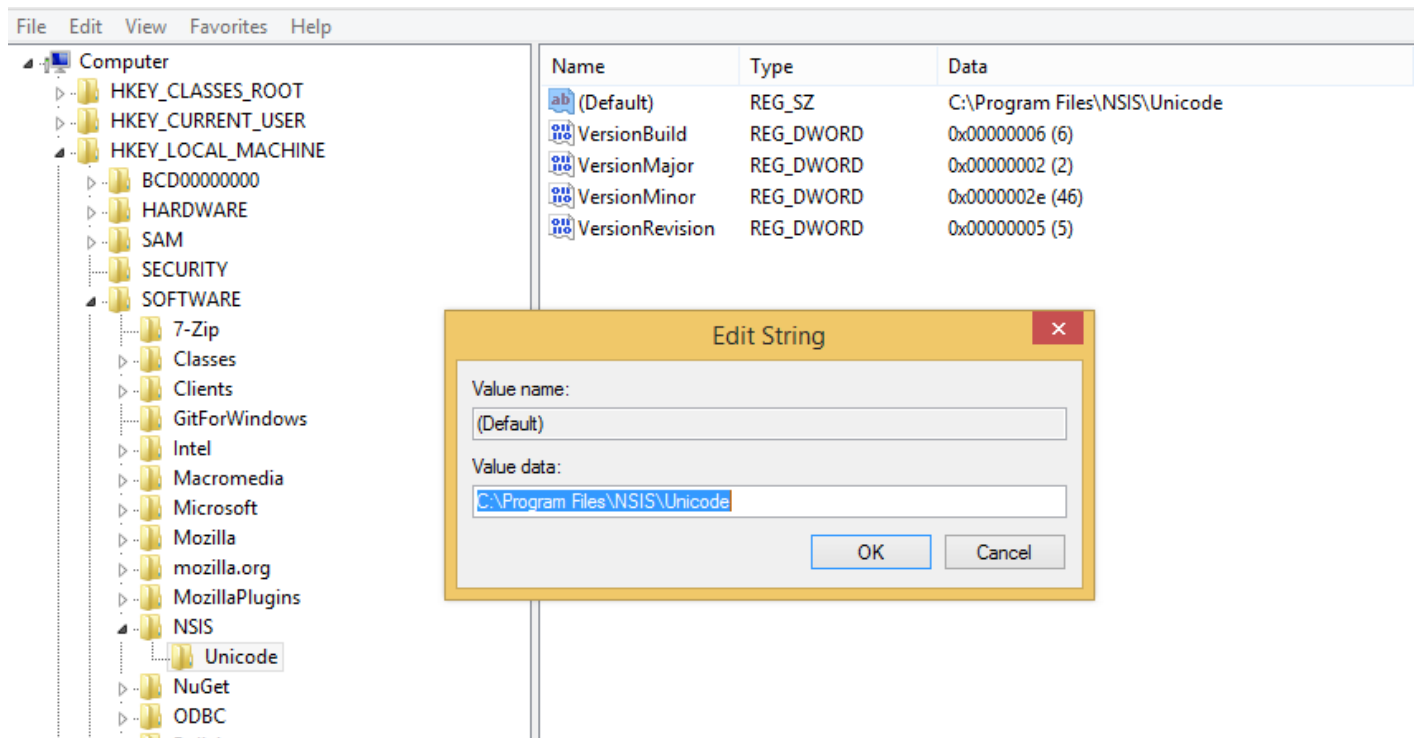
- Open a 'Command Prompt' and type and run 'regedit'. This starts the 'Registry Editor'.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Mike>regedit
C:\Users\Mike>
```

- Navigate to '**HKEY_LOCAL_MACHINE\SOFTWARE\NSIS\Unicode**'. Double-click on the 'Default' line.

Note: If you are running a 64-bit machine (x64) the key is located in 'HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432node\NSIS\Unicode'

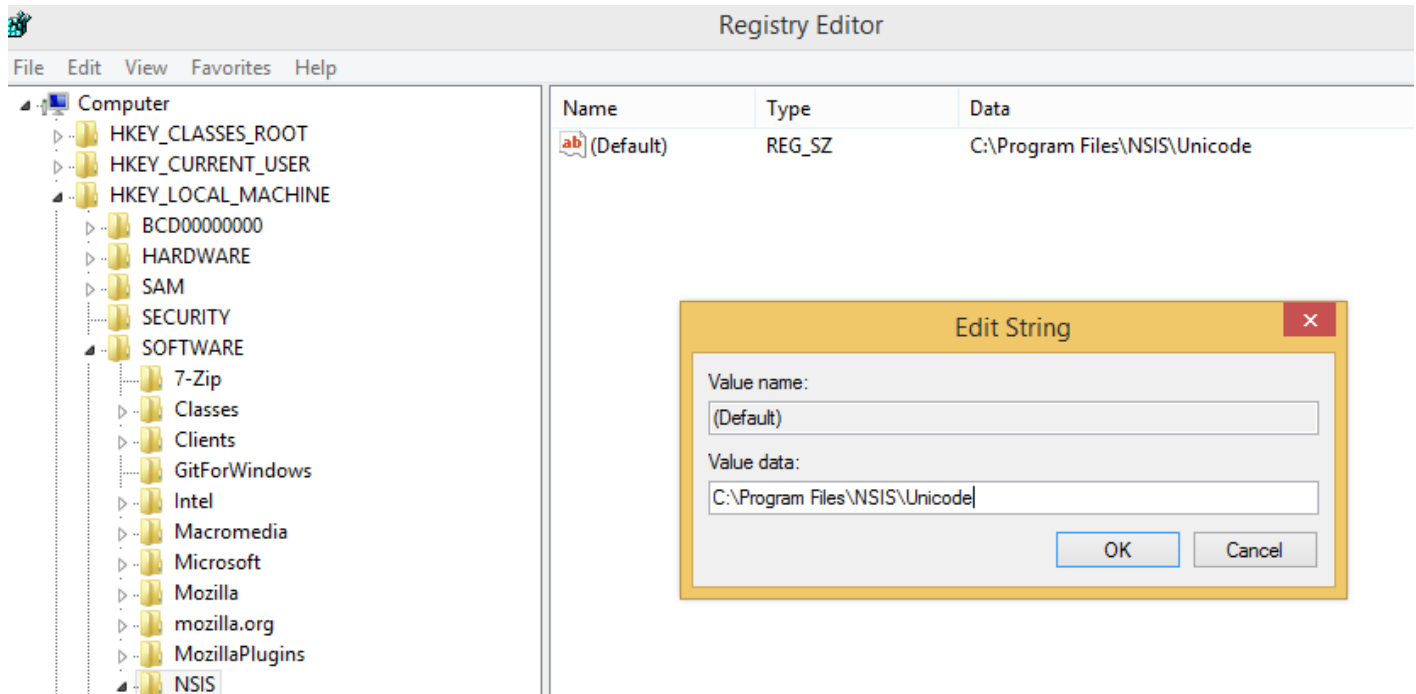


- Copy the value (The installation path of NSIS).
- Navigate to '**HKEY_LOCAL_MACHINE\SOFTWARE\NSIS**'.

Note: If you are running a 64-bit machine (x64) the location is 'HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432node\NSIS'.

Name	Type	Data
(Default)	REG_SZ	(value not set)

- Double-click on the 'Default' line and paste the install path into 'Value data'.



To make the installer package use proper language name translations, it is necessary to modify file '**C:\Program Files\NSIS\Unicode\Contrib\Language files\Norwegian.nsh**' and change the line

```
!insertmacro LANGFILE "Norwegian" "Norwegian"
```

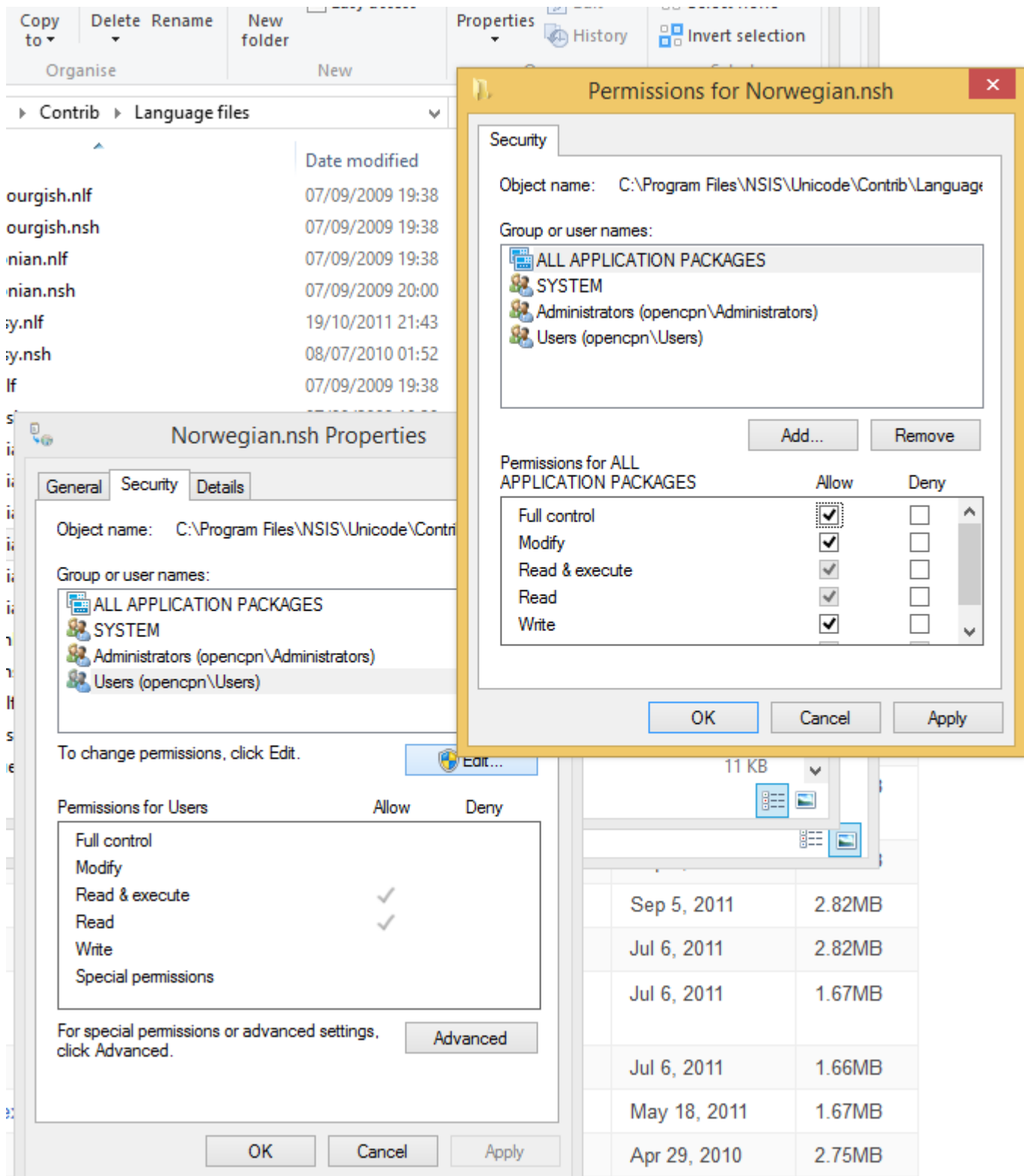
to

```
!insertmacro LANGFILE "Norwegian" "Norsk"
```

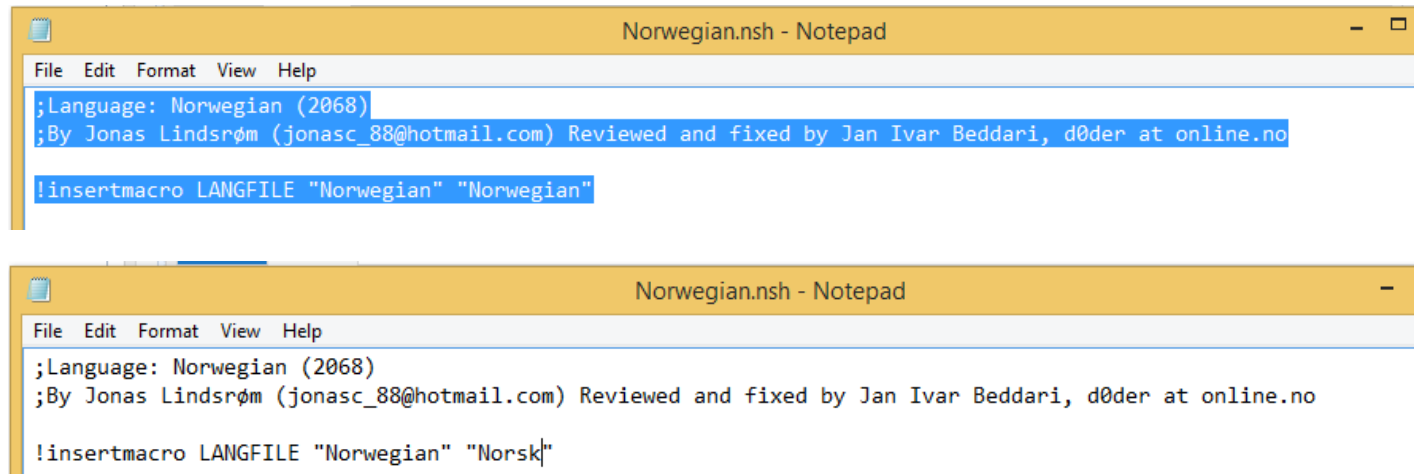
The '**C:\Program Files\NSIS\Unicode\Contrib\Language files\Norwegian.nsh**' is edited.

Due to 'Security' you will need to add 'Write' permission to this file. Without this you will not be able to save the changes.

- With 'Explorer' navigate to '**C:\Program Files\NSIS\Unicode\Contrib\Language files**'.
- Right-click on the file '**Norwegian.nsh**'.
- In 'Properties', 'Security' tab, press the 'Edit' button.

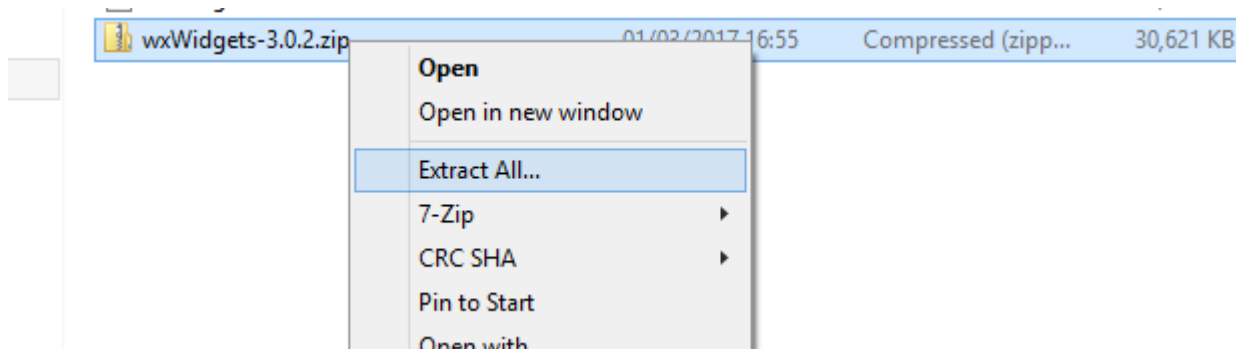


- Select 'Users' and tick all the 'Allow' boxes.
- The '**C:\Program Files\NSIS\Unicode\Contrib\Language files\Norwegian.nsh**' can now be opened with 'WordPad' or 'Notepad' and the changes made and saved.

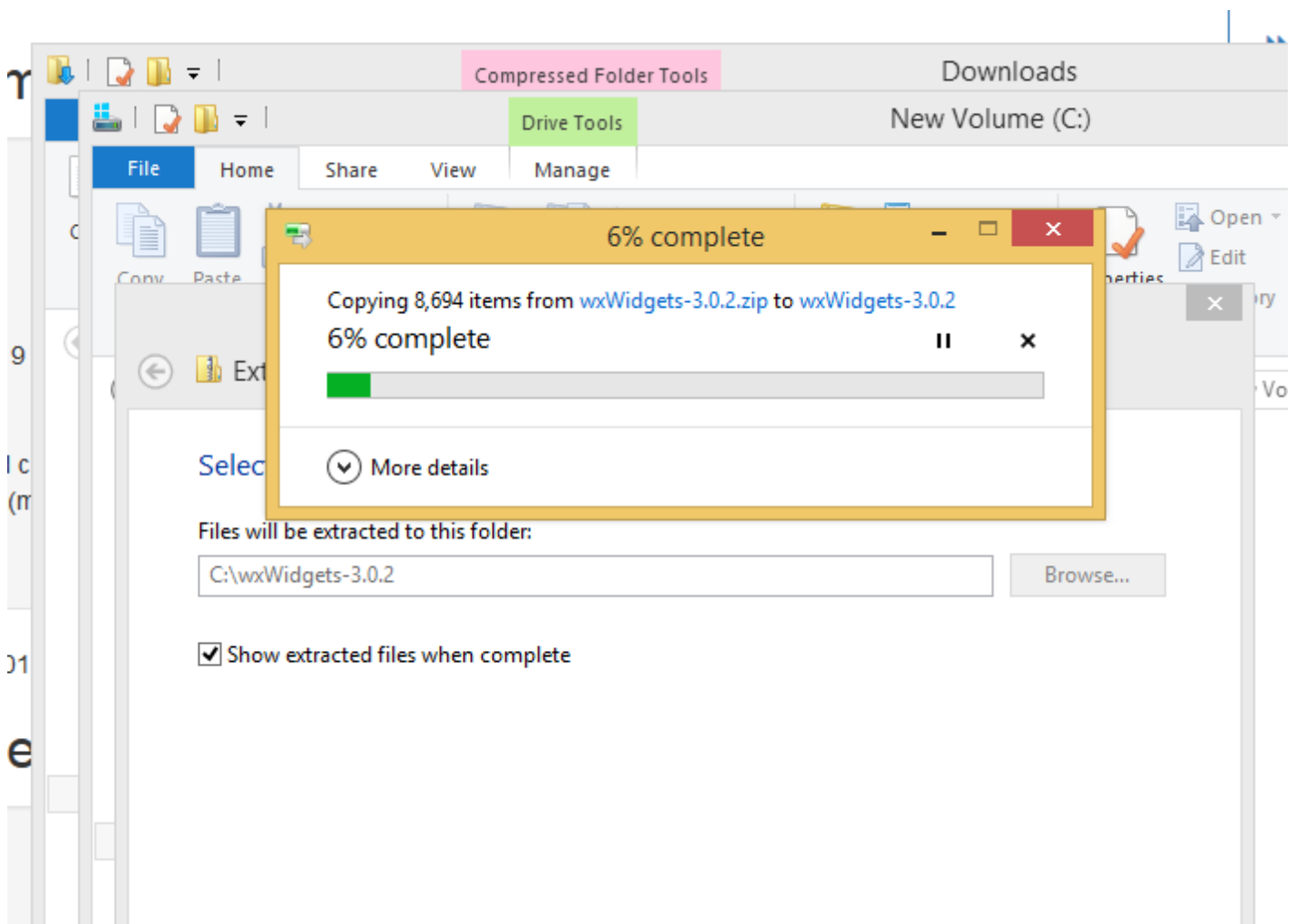


Compiling wxWidgets 3.0.2

- Download the '3.0.2 release' as a 'zip' file from <http://wxwidgets.org/downloads/>
- Navigate to the folder where you downloaded the zip.
- Right-click on the file '**wxWidgets-3.0.2.zip**' and select the menu option 'Extract All...'.

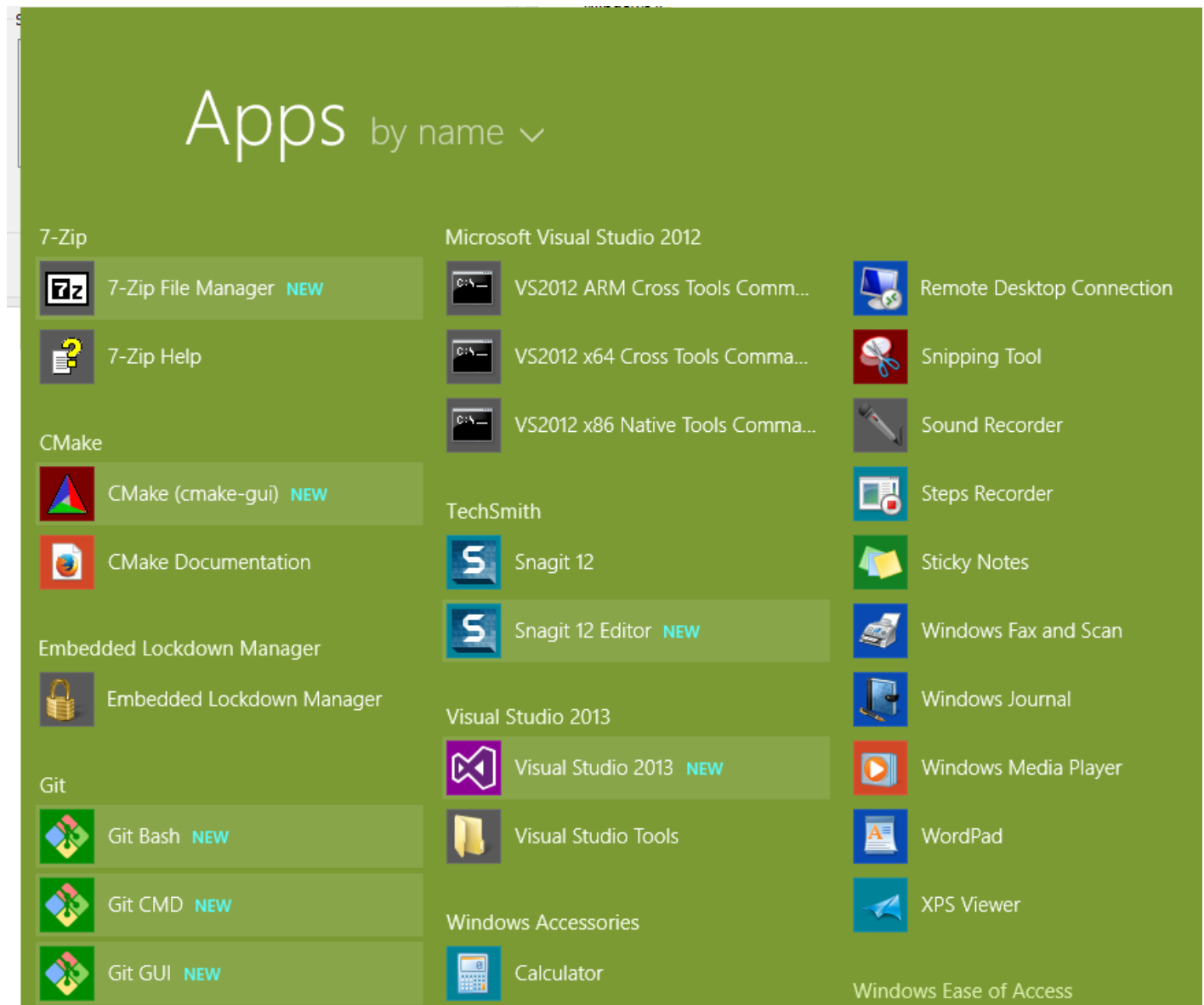


- Edit the folder for the Destination to read 'C:\wxWidgets-3.0.2' and press the 'Extract' button.

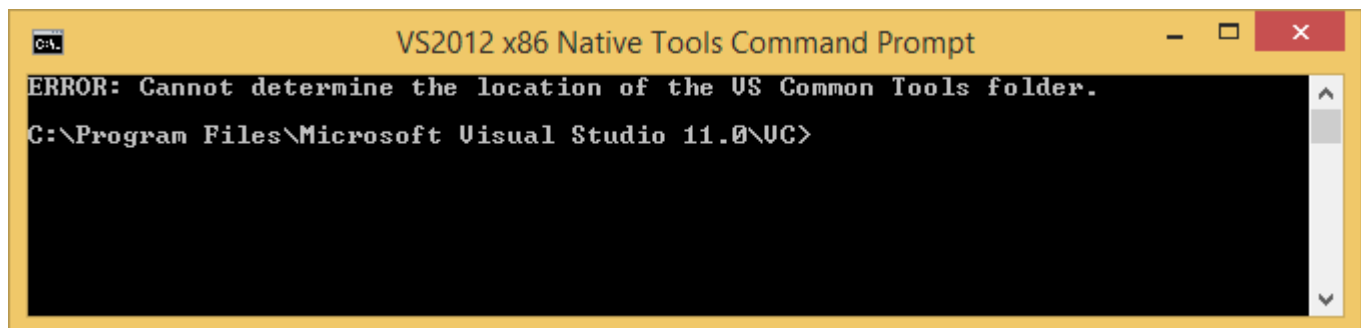


Start the 'VS2013 x86 Native Tools Command Prompt'

Note: This can be confusing!!! Even though we have installed 'Visual Studio Community 2013' when you search the 'Program Files' folder you will see 'Microsoft Visual Studio 12' but not 'Microsoft Visual Studio 13'. When you look at the installed 'Apps' you will see this:

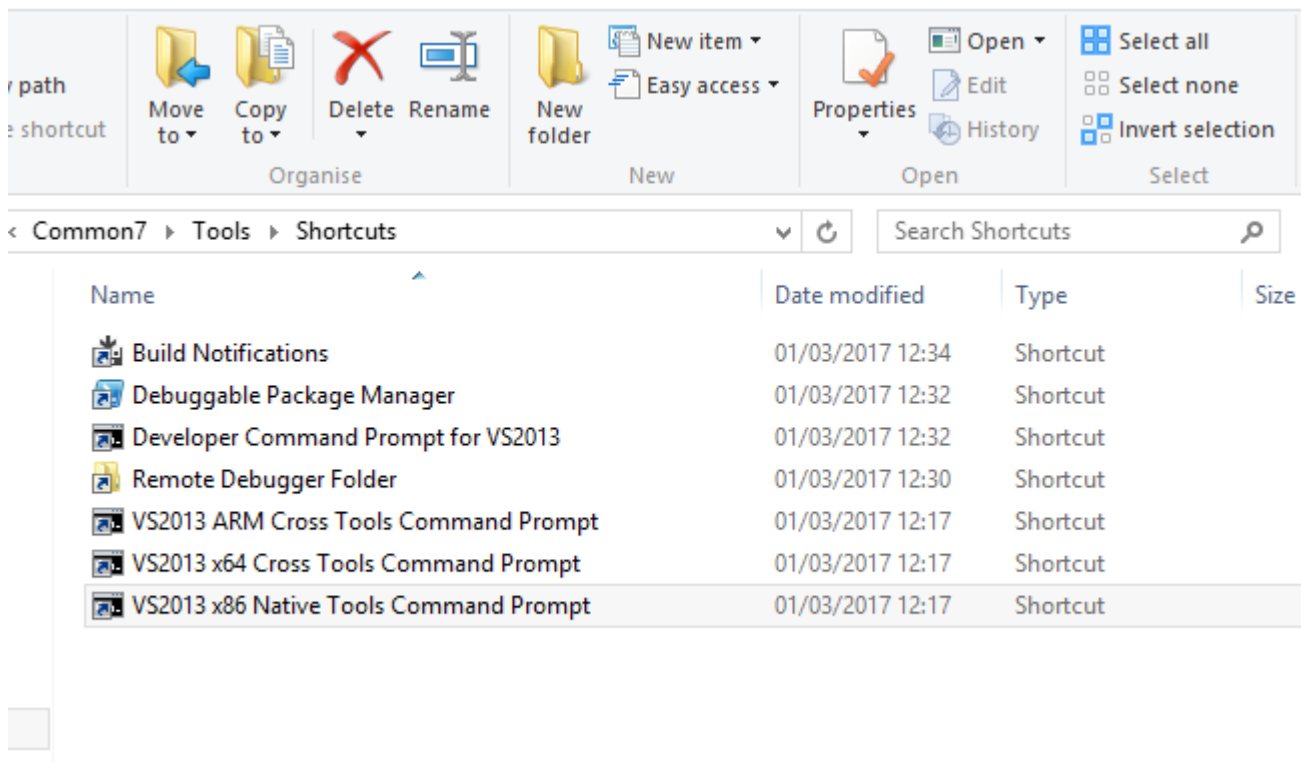


- If you start the command prompt from 'Apps' **DO NOT** use the 'Microsoft Visual Studio 2012 - VS2012 x86 Native Tools Command Prompt'. An error message will appear:



- Instead ... press the button 'Visual Studio Tools' under the heading 'Visual Studio 2013'.

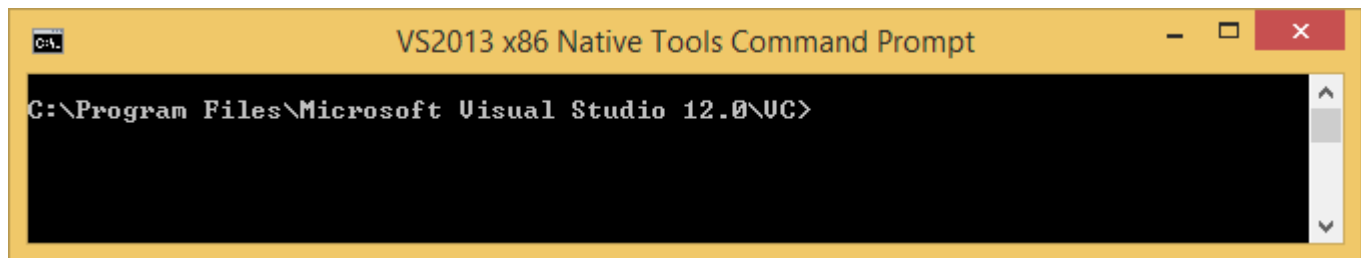
This will open a folder of shortcuts.



You are in the folder 'C:\Program Files\Microsoft Visual Studio 12.0\Common7\Tools\Shortcuts'

- Double-click the 'VS2013 x86 Native Tools Command Prompt' shortcut and a 'Terminal' window will appear

Note: On a 64 bit machine the prompt is 'VS2013 x64 Command Prompt'



Title is 'VS2013', the text reads 'Visual Studio 12'. You get the picture!!!

- cd to '**C:\wxWidgets-3.0.2\build\msw**'.

An easy way to enter the correct folder (cd) for 'wxWidgets 3.0.2\build\msw' is to navigate to that folder with 'Explorer'.

- On the line showing the folders right-click and select 'Copy address as text'.
- In the Terminal window type 'cd ' (that is 'cd' plus a space'). Right-click and 'Paste'.

Build both **release** and **debug** configurations, which will be compatible with Windows XP.

RELEASE VERSION


```
nmake -f makefile.vc BUILD=release SHARED=1 CFLAGS=/D_USING_V120_SDK71_
CXXFLAGS=/D_USING_V120_SDK71_
```

DEBUG VERSION

```
nmake -f makefile.vc BUILD=debug SHARED=1 CFLAGS=/D_USING_V120_SDK71_
CXXFLAGS=/D_USING_V120_SDK71_
```

- For the 'Release' version copy the line starting 'nmake' into the Terminal window and press 'Enter'. Wait until the building has finished. This could take some time.
- Repeat the process for the 'Debug' version.

```

.\..\src\zlib\I..\..\src\regex\I..\..\src\expat\lib\DWXUSINGDLL\DWXMAKINGDLL
L_GL\GR\EHsc\Yu"wx\wxprec.h" /Fp"vc_mswudll\wxprec_gldll.pch" /D_USING_V120
SDK71_ /Ycwx\wxprec.h .....\src\common\dummy.cpp
dummy.cpp
cl /c /nologo /TP /Fovc_mswudll\gldll_glcmm.obj /MD /DWIN32 /Zi /Fd..\
..\lib\vc_dll\wxmsw30u_gl_vc_custom.pdb /O2 /D_CRT_SECURE_NO_DEPRECATED=1 /D_C
RT_NON_CONFORMING_SWPRINTFS=1 /D_SCL_SECURE_NO_WARNINGS=1 /D_NO_VC_CRTDBG_
/D_WXMSW_ /DNDEBUG /D_UNICODE /I..\..\lib\vc_dll\mswu /I..\..\include
/W4 /DWXBUILDING /I..\..\src\tiff\libtiff /I..\..\src\jpeg /I..\..\src\png /I
..\..\src\zlib\I..\..\src\regex\I..\..\src\expat\lib\DWXUSINGDLL\DWXMAKINGDLL
L_GL\GR\EHsc\Yu"wx\wxprec.h" /Fp"vc_mswudll\wxprec_gldll.pch" /D_USING_V120
SDK71_ .....\src\common\glcmm.cpp
glcmm.cpp
cl /c /nologo /TP /Fovc_mswudll\gldll_glcmm.obj /MD /DWIN32 /Zi /Fd
..\..\lib\vc_dll\wxmsw30u_gl_vc_custom.pdb /O2 /D_CRT_SECURE_NO_DEPRECATED=1 /
D_CRT_NON_CONFORMING_SWPRINTFS=1 /D_SCL_SECURE_NO_WARNINGS=1 /D_NO_VC_CRTDBG_
/D_WXMSW_ /DNDEBUG /D_UNICODE /I..\..\lib\vc_dll\mswu /I..\..\include
/W4 /DWXBUILDING /I..\..\src\tiff\libtiff /I..\..\src\jpeg /I..\..\src\png /I
..\..\src\zlib\I..\..\src\regex\I..\..\src\expat\lib\DWXUSINGDLL\DWXMAKIN
GDLL_GL\GR\EHsc\Yu"wx\wxprec.h" /Fp"vc_mswudll\wxprec_gldll.pch" /D_USING_U
120_SDK71_ .....\src\msw\glcmm.cpp
glcmm.cpp
rc /fvc_mswudll\gldll_version.res /d WIN32 /d _CRT_SECURE_NO_DEPRECATED
E=1 /d _CRT_NON_CONFORMING_SWPRINTFS=1 /d _SCL_SECURE_NO_WARNINGS=1 /d _NO_VC_C
RTDBG_ /d _WXMSW_ /d NDEBUG /d UNICODE /i .....\lib\vc_dll\mswu /i ..
\..\include /d WXBUILDING /d WXDLLNAME=wxmsw30u_gl_vc_custom /i .....\src\tiff
\libtiff /i .....\src\jpeg /i .....\src\png /i .....\src\zlib /i .....\src\regex
/i .....\src\expat\lib /d WXUSINGDLL /d WXMAKINGDLL_GL .....\src\msw\version.rc
Microsoft (R) Windows (R) Resource Compiler Version 6.3.9600.17298
Copyright (C) Microsoft Corporation. All rights reserved.

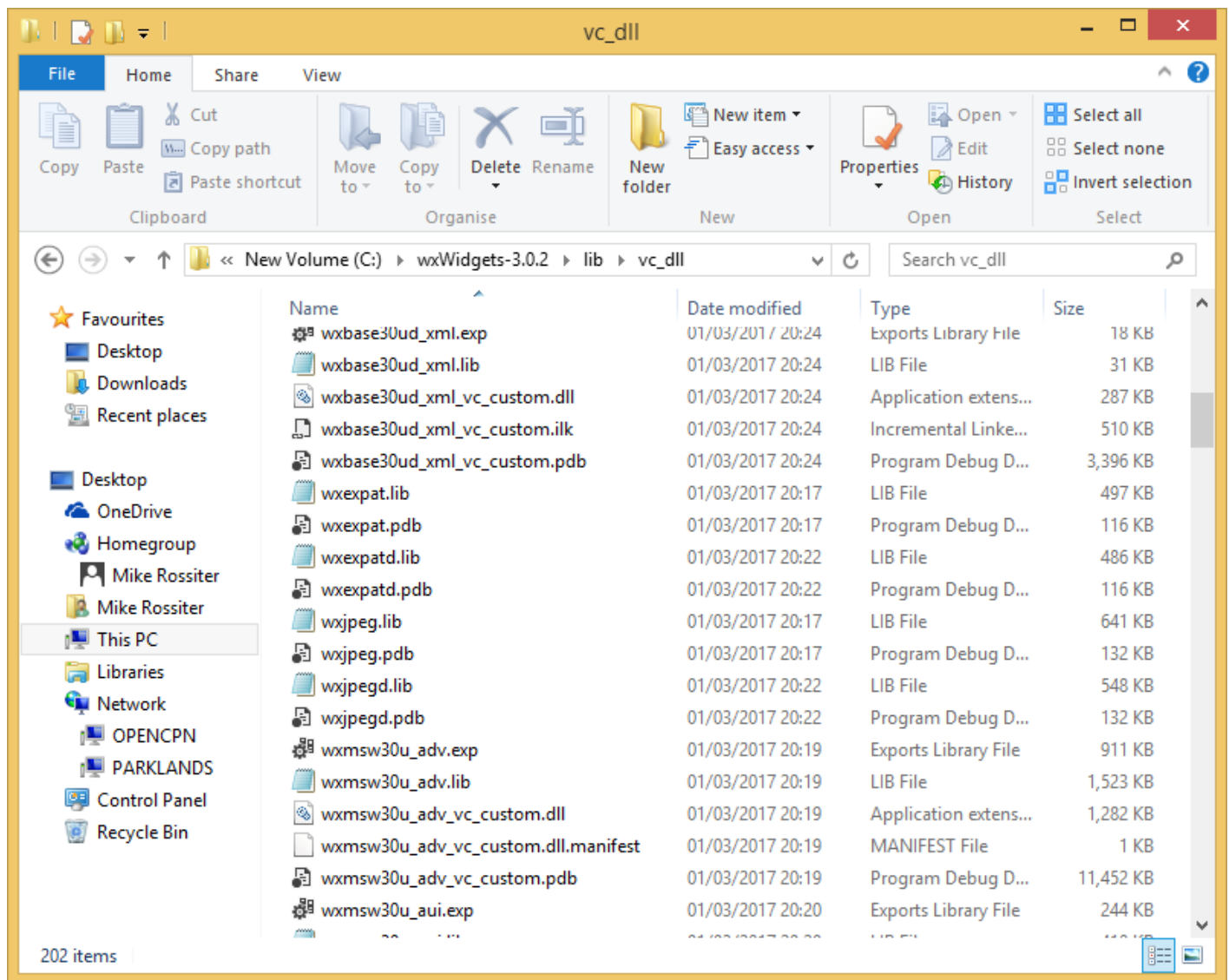
link /DLL /NOLOGO /OUT:..\..\lib\vc_dll\wxmsw30u_gl_vc_custom.dll /DEBU
G /pdb:"..\..\lib\vc_dll\wxmsw30u_gl_vc_custom.pdb" /opt:ref /opt:icf /LIBPATH
..\..\lib\vc_dll @C:\Users\Mike\AppData\Local\Temp\nm31C7.tmp
Creating library .....\lib\vc_dll\wxmsw30u_gl.lib and object .....\lib\vc_dll
\wxmsw30u_gl.exp
c:\wxWidgets-3.0.2\build\msw>

```

- Close the 'Terminal' window.

You will find that a number of files have been made in the folder '**c:\wxWidgets-3.0.2\lib\vc-dll**'.

Some of the filenames start 'wxmsw30u' and others 'wxmsw30ud' corresponding to the 'Release' and 'Debug' versions.

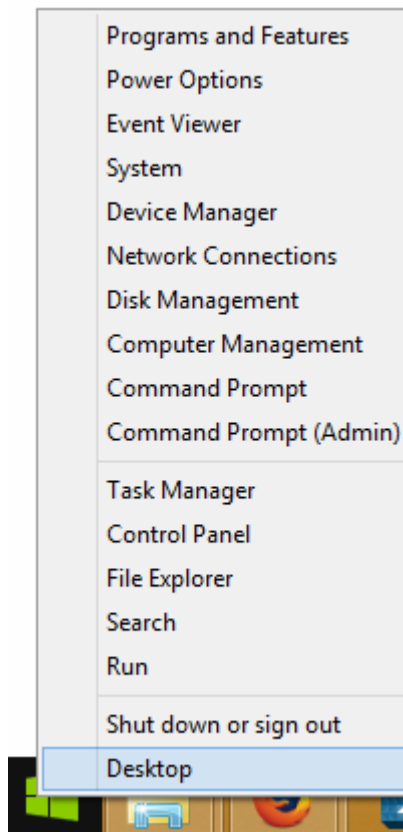


This completes the preparations for building the OpenCPN program.

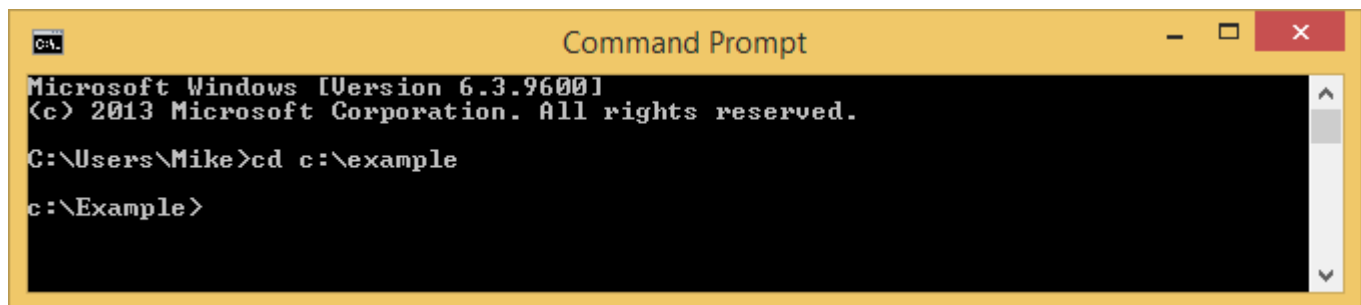
Building OpenCPN

Get the OpenCPN source

- Make a folder to store your OpenCPN source code files. In this guide I am going to call it 'Example' in the root folder, i.e. '**C:\Example**'.
- Start a 'Command Prompt' (Any prompt will work - just right-click on your Window icon) and select 'Command Prompt'. A Terminal window will appear.



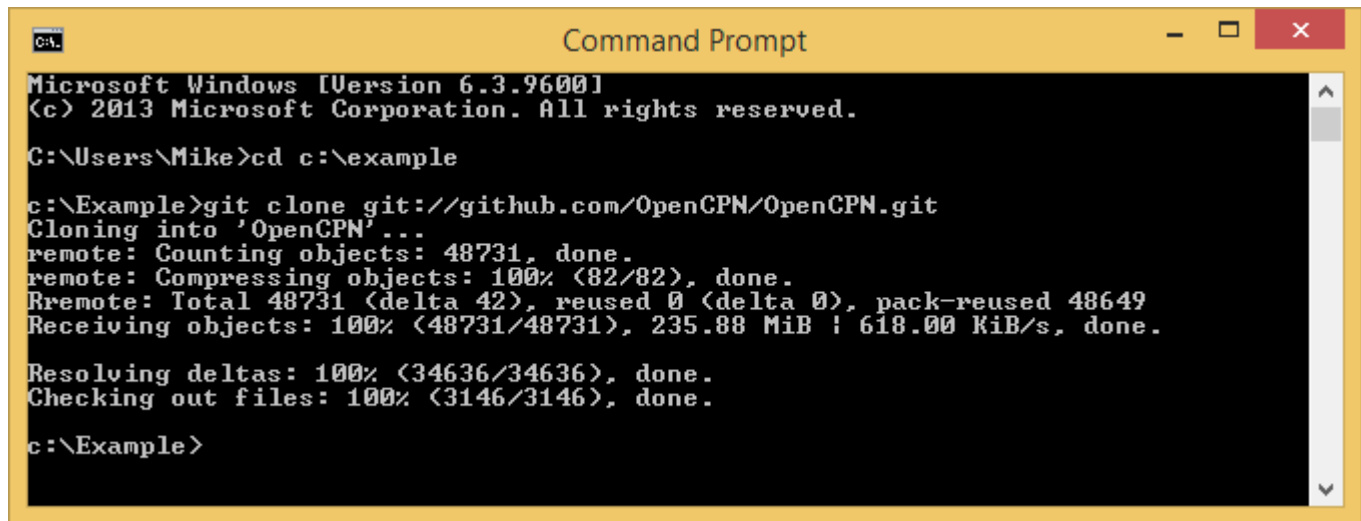
- CD to the 'Example' folder. (Type 'cd C:\Example' and press 'Enter').



- Type this text into the Terminal window and press 'Enter'.

```
git clone git://github.com/OpenCPN/OpenCPN.git
```

- This will download the latest Beta code.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Mike>cd c:\example

c:\Example>git clone git://github.com/OpenCPN/OpenCPN.git
Cloning into 'OpenCPN'...
remote: Counting objects: 48731, done.
remote: Compressing objects: 100% (82/82), done.
remote: Total 48731 (delta 42), reused 0 (delta 0), pack-reused 48649
Receiving objects: 100% (48731/48731), 235.88 MiB | 618.00 KiB/s, done.
Resolving deltas: 100% (34636/34636), done.
Checking out files: 100% (3146/3146), done.

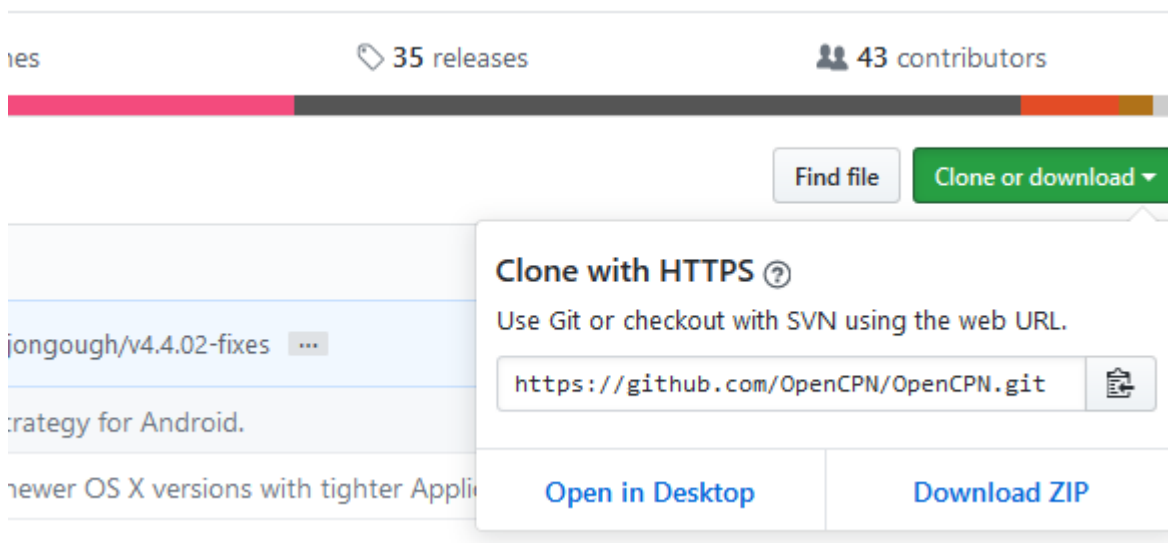
c:\Example>
```

- The files/folders for building 'OpenCPN' will be placed in the folder 'C:\Example\OpenCPN'
- If you are happy to work with the Beta version source code ... move on to the [next section](#).

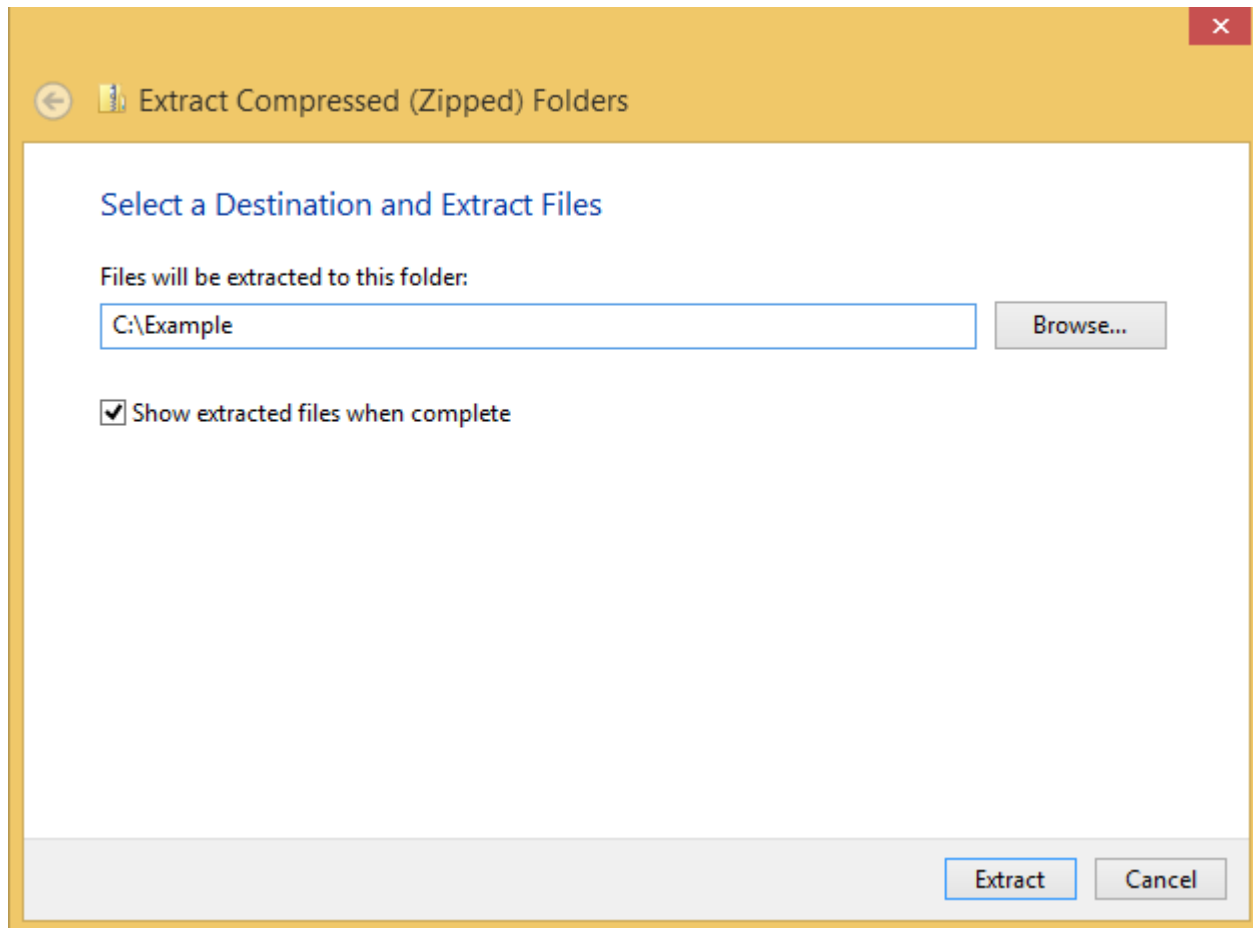
Note: If you want the source code for the latest stable release (4.4.0 at time of writing) you need to locate that source on GitHub.com:

<https://github.com/OpenCPN/OpenCPN/tree/v4.4.0>

- Press the 'Clone or download' button. **DO NOT** use the text for 'git clone' or you will get the Beta version. Instead ...
- Select 'Download Zip' and get the zip file.



- Extract the files to the folder 'C:\Example'

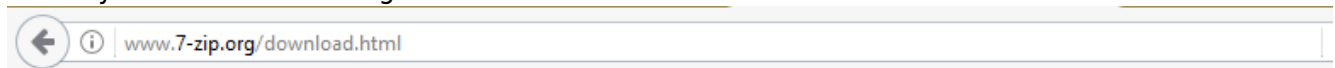


- The files/folders for building 'OpenCPN' will be placed in the folder 'C:\Example\OpenCPN-4.4.0'
- These are the files that are going to be used for this workflow. This folder is renamed 'C:\Example\OpenCPN' to make the process steps read in the same way as for the files from 'git clone' (the Beta version).

Get the binary dependency files

Sorry, this needs another program - **7Zip**.

- Get the installer from [here](#).
- 7z files are compressed files making them smaller and faster to download. The '7Zip' program allows you to extract the original files.

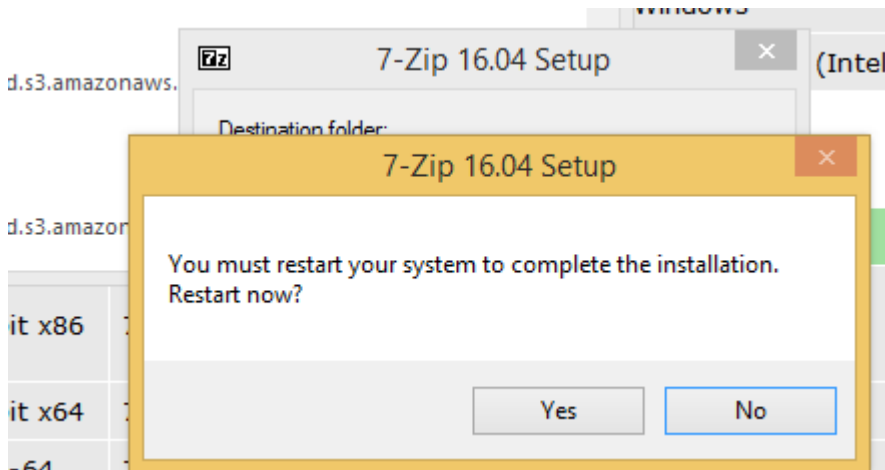


Download

Download 7-Zip 16.04 (2016-10-04) for Windows:

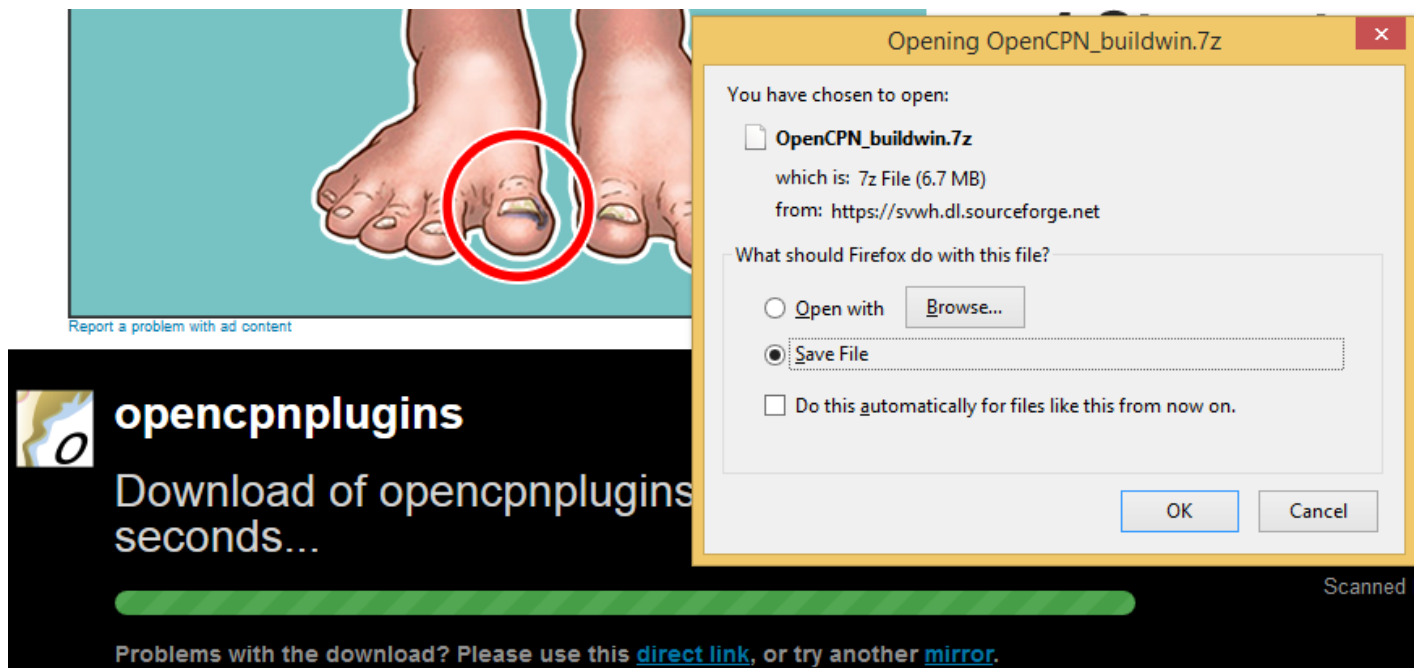
Link	Type	Windows	Description
Download	.exe	32-bit x86	7-Zip for 32-bit Windows

- After the download double-click the '7z....exe' file to carry out the installation.
- You need to restart the computer.

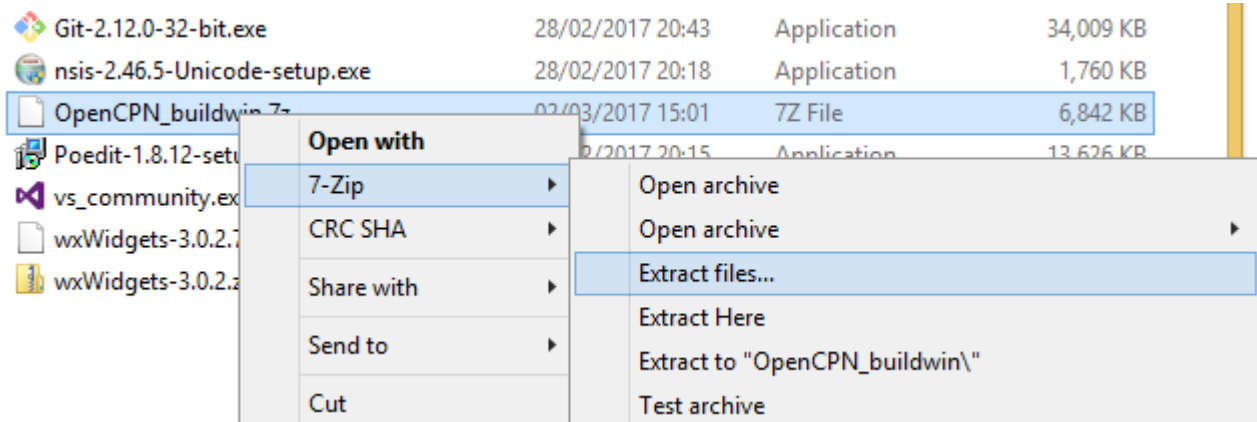


'7z' files can now be opened and extracted with this program.

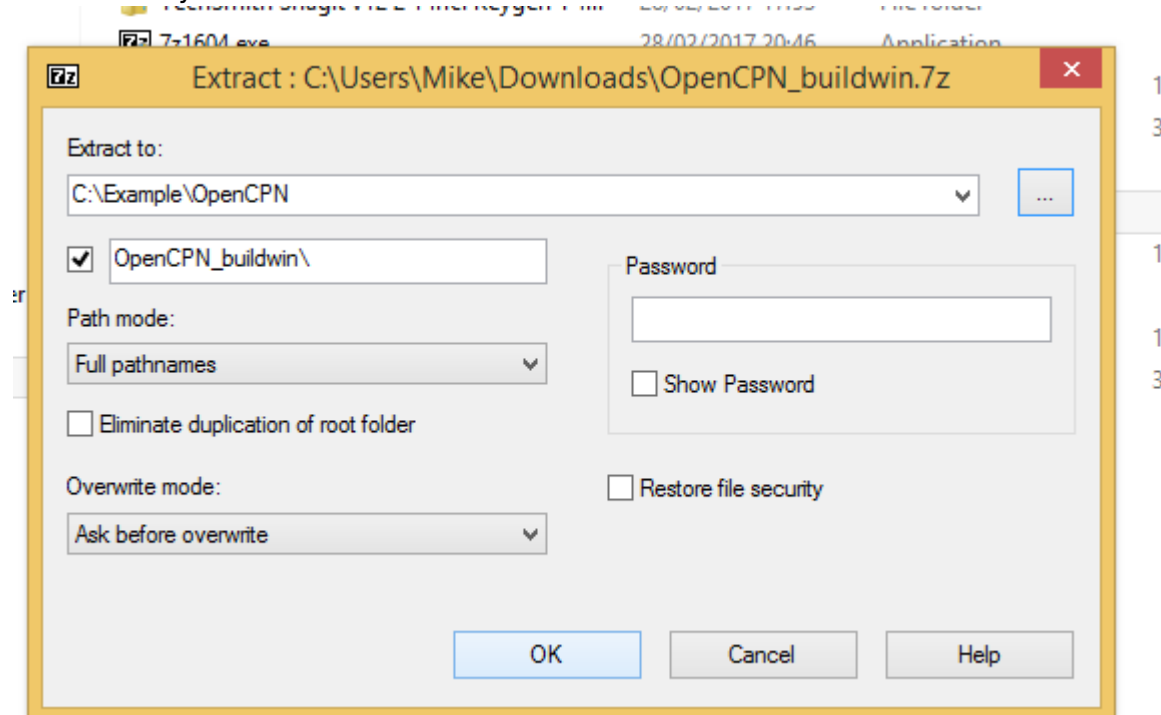
- Download [OpenCPN_buildwin.7z](#)



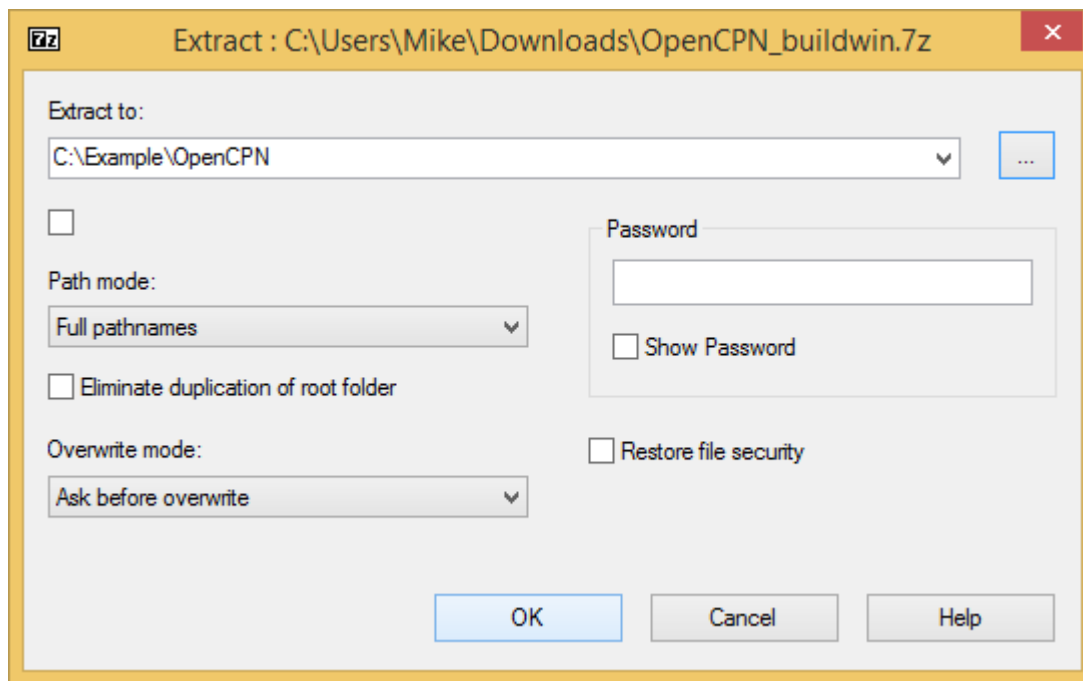
- Right-click on this file in 'Windows Explorer'. Select the option '7-Zip', 'Extract Files'.



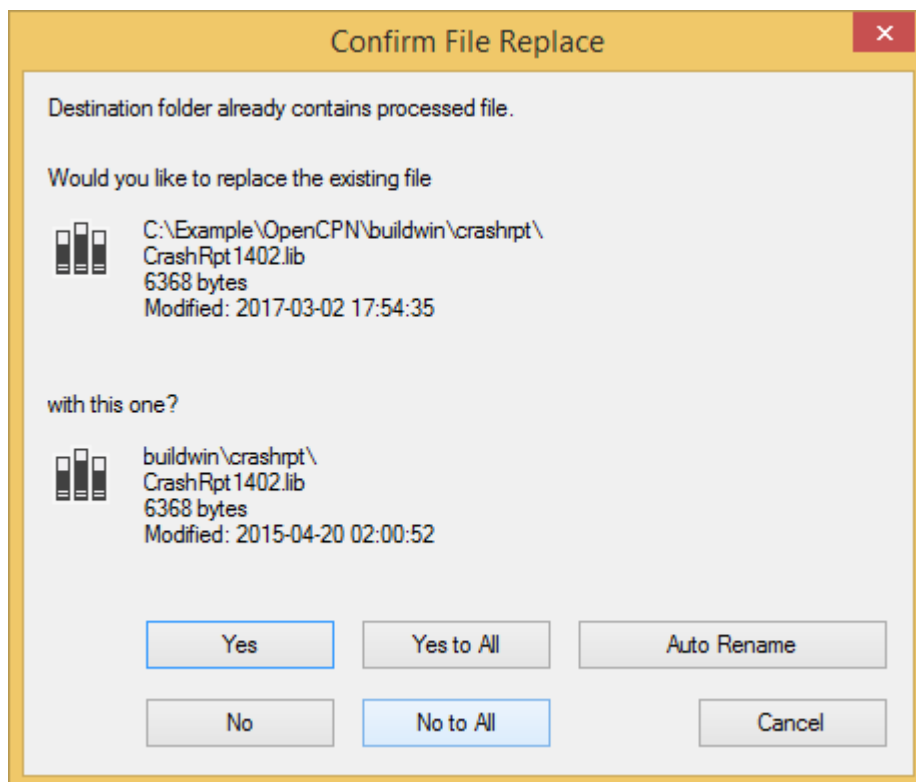
- Select the folder 'C:\Example\OpenCPN'. The files and folders from the 7z file will be placed under that directory



- Uncheck the box next to the text 'OpenCPN_buildwin'.



- Use the 'No to All' button to avoid overwriting files in the source.



- This will add extra files in the folder 'C:\Example\OpenCPN\buildwin' that are needed for the build.

Make a Visual Studio solution for building OpenCPN

- Start the VS2013 x86 Native Tools Command Prompt (A reminder is [here](#))

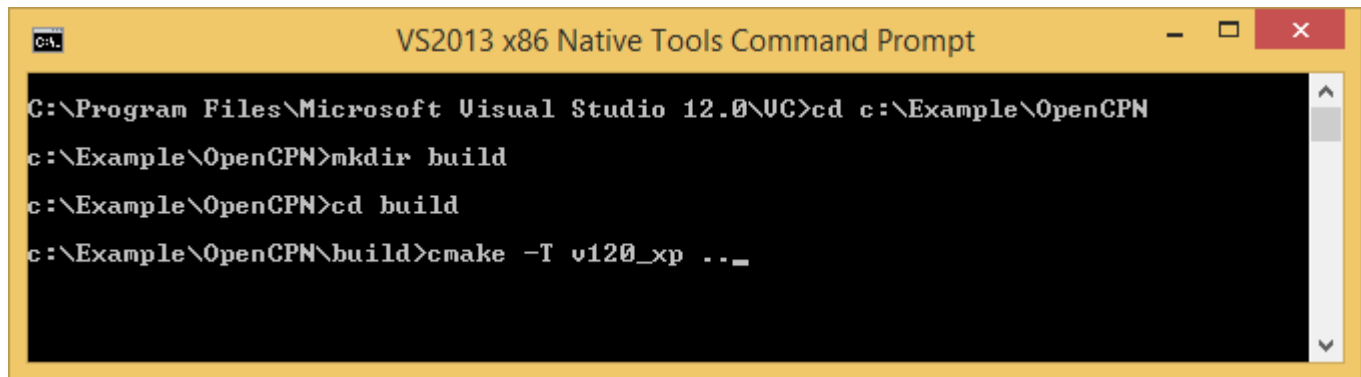
[The command prompt shortcut is in the folder 'C:\Program Files\Microsoft Visual Studio 12.0\Common7\Tools\Shortcuts']

- Change Directory [cd] into 'C:\Example\OpenCPN'.
- Create a folder named "build" under this topmost source folder.

```
mkdir build
```

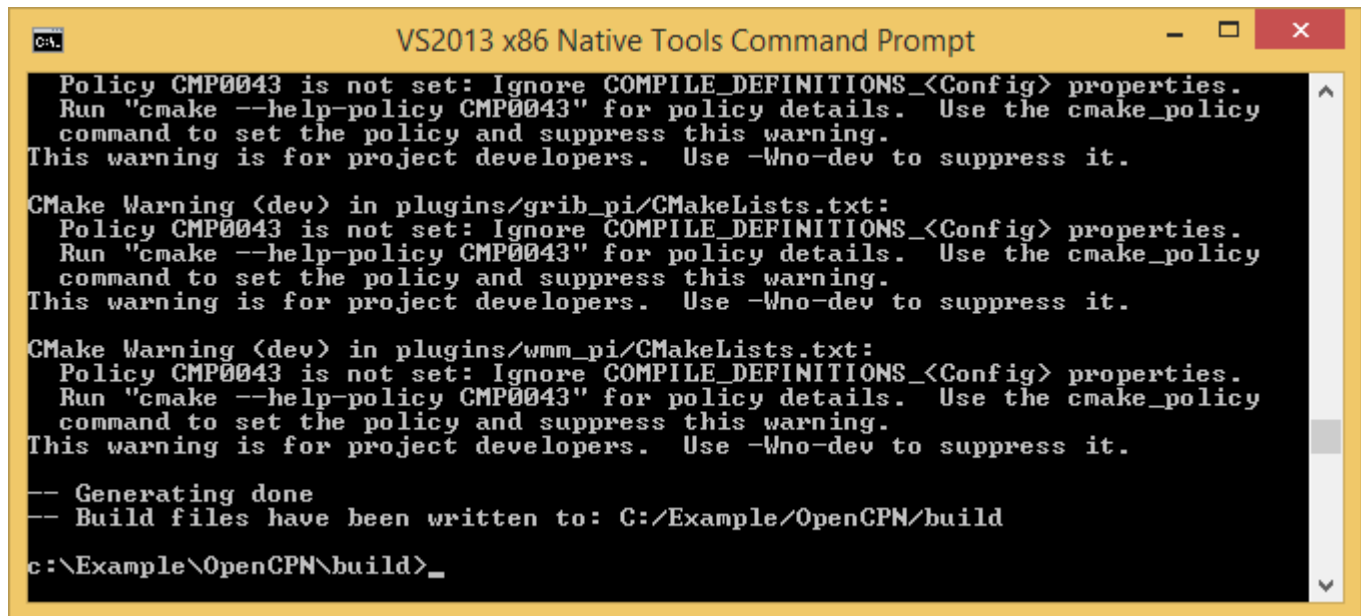
- cd to the "build" folder and then issue the cmake command shown.

```
cd build
cmake -T v120_xp ..
```



```
VS2013 x86 Native Tools Command Prompt

C:\Program Files\Microsoft Visual Studio 12.0\VC>cd c:\Example\OpenCPN
c:\Example\OpenCPN>mkdir build
c:\Example\OpenCPN>cd build
c:\Example\OpenCPN\build>cmake -T v120_xp .._
```



```
VS2013 x86 Native Tools Command Prompt

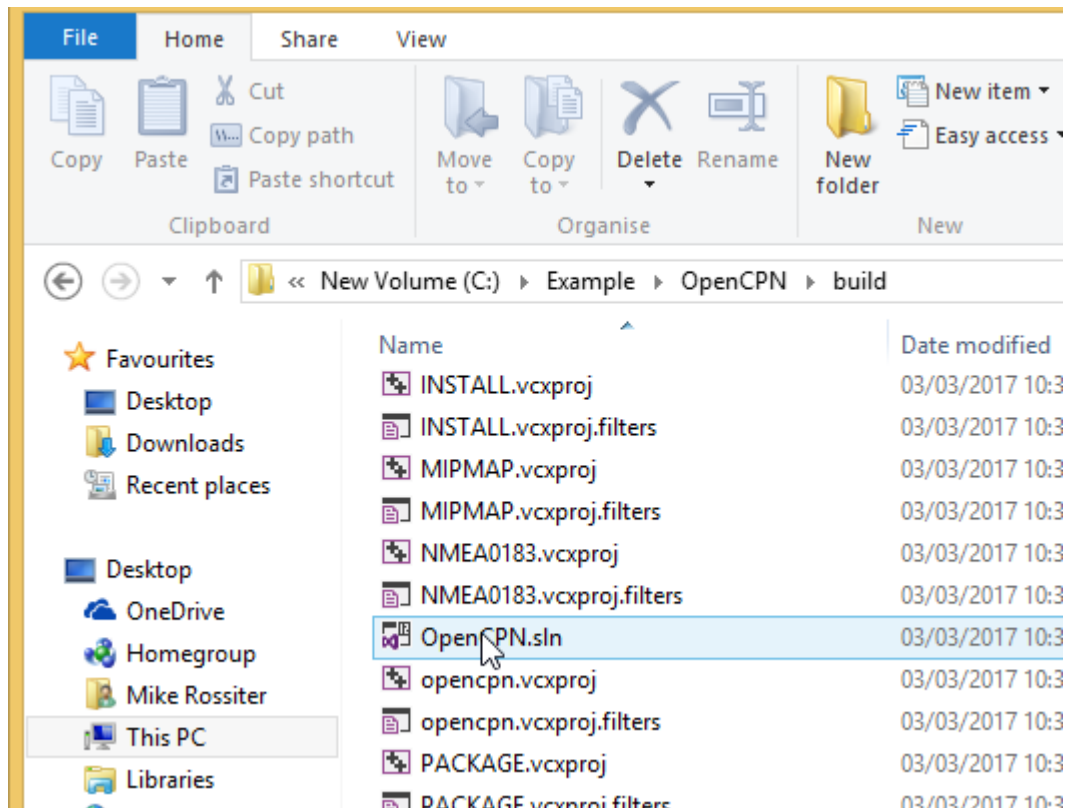
Policy CMP0043 is not set: Ignore COMPILE_DEFINITIONS_<Config> properties.
Run "cmake --help-policy CMP0043" for policy details. Use the cmake_policy
command to set the policy and suppress this warning.
This warning is for project developers. Use -Wno-dev to suppress it.

CMake Warning (dev) in plugins/grib_pi/CMakeLists.txt:
Policy CMP0043 is not set: Ignore COMPILE_DEFINITIONS_<Config> properties.
Run "cmake --help-policy CMP0043" for policy details. Use the cmake_policy
command to set the policy and suppress this warning.
This warning is for project developers. Use -Wno-dev to suppress it.

CMake Warning (dev) in plugins/wmm_pi/CMakeLists.txt:
Policy CMP0043 is not set: Ignore COMPILE_DEFINITIONS_<Config> properties.
Run "cmake --help-policy CMP0043" for policy details. Use the cmake_policy
command to set the policy and suppress this warning.
This warning is for project developers. Use -Wno-dev to suppress it.

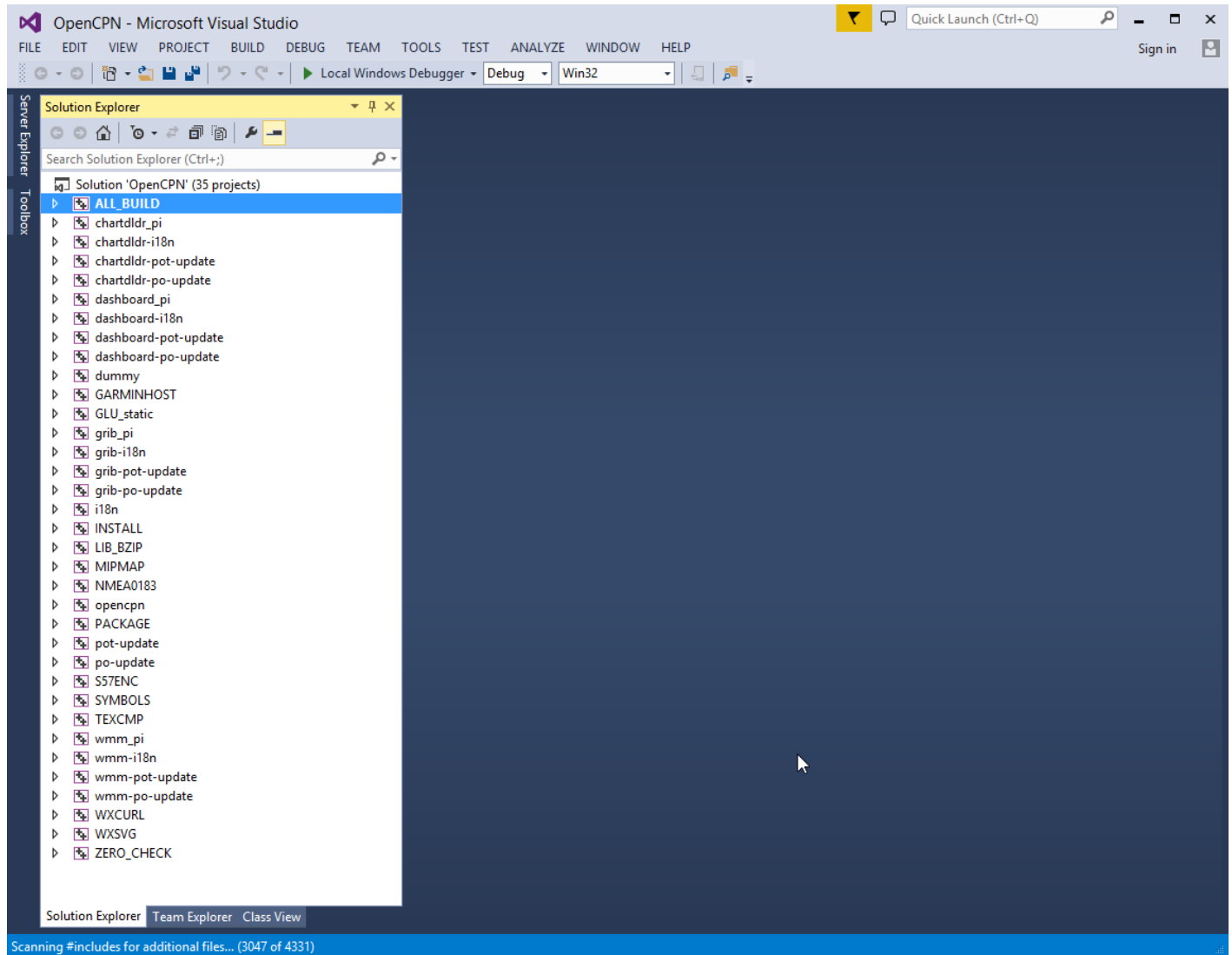
-- Generating done
-- Build files have been written to: C:/Example/OpenCPN/build
c:\Example\OpenCPN\build>_
```

- * Close the Terminal window.
- * This has created the Visual Studio solution file 'OpenCPN.sln'.

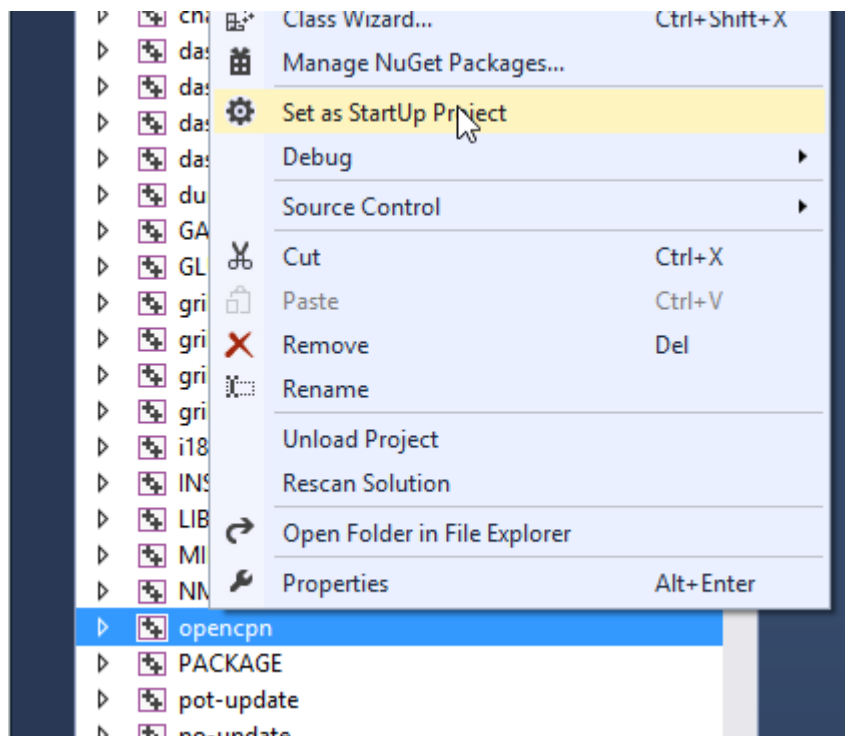


Build OpenCPN

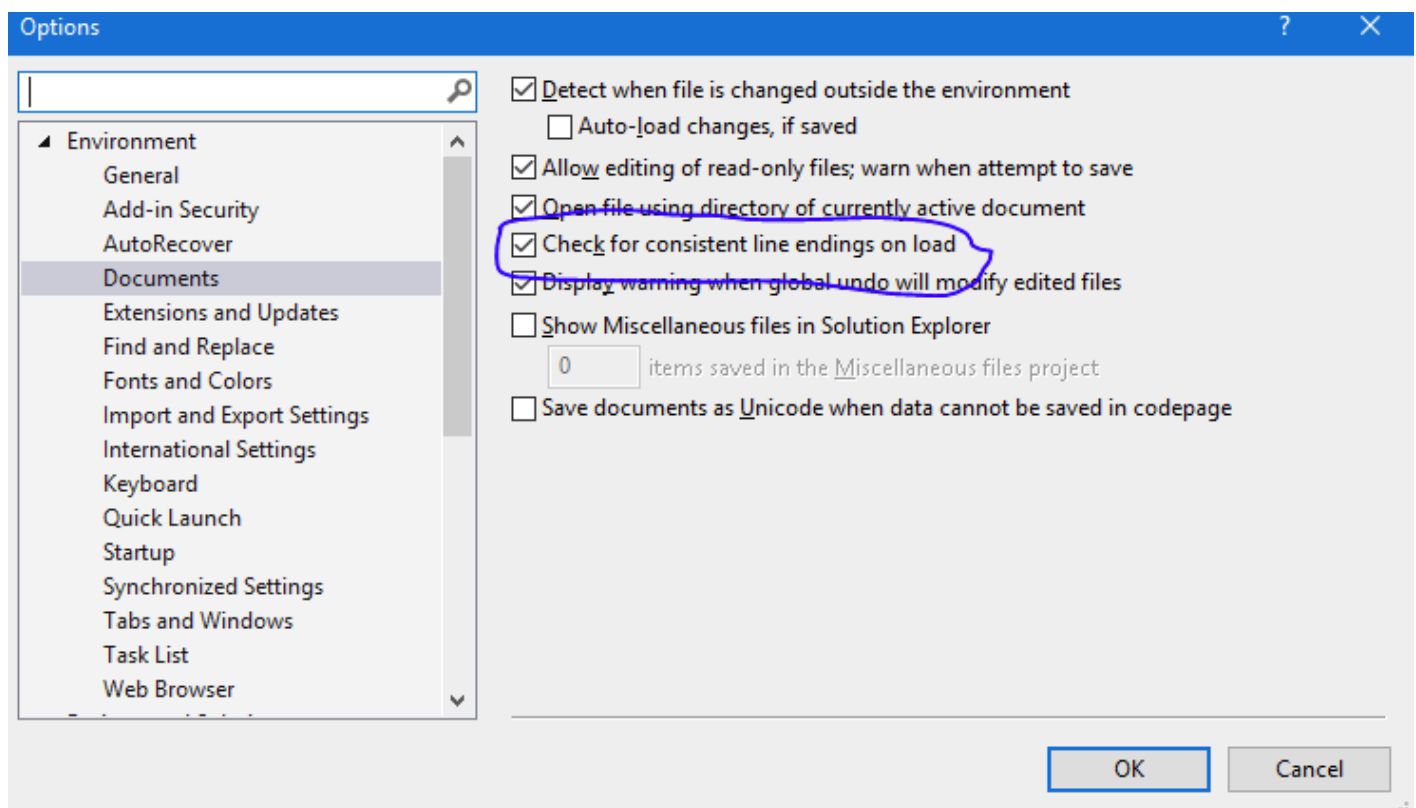
- Open the '.sln' file with the program Visual Studio Community 2013 (you can double-click the file name)
- The program window should look like this:

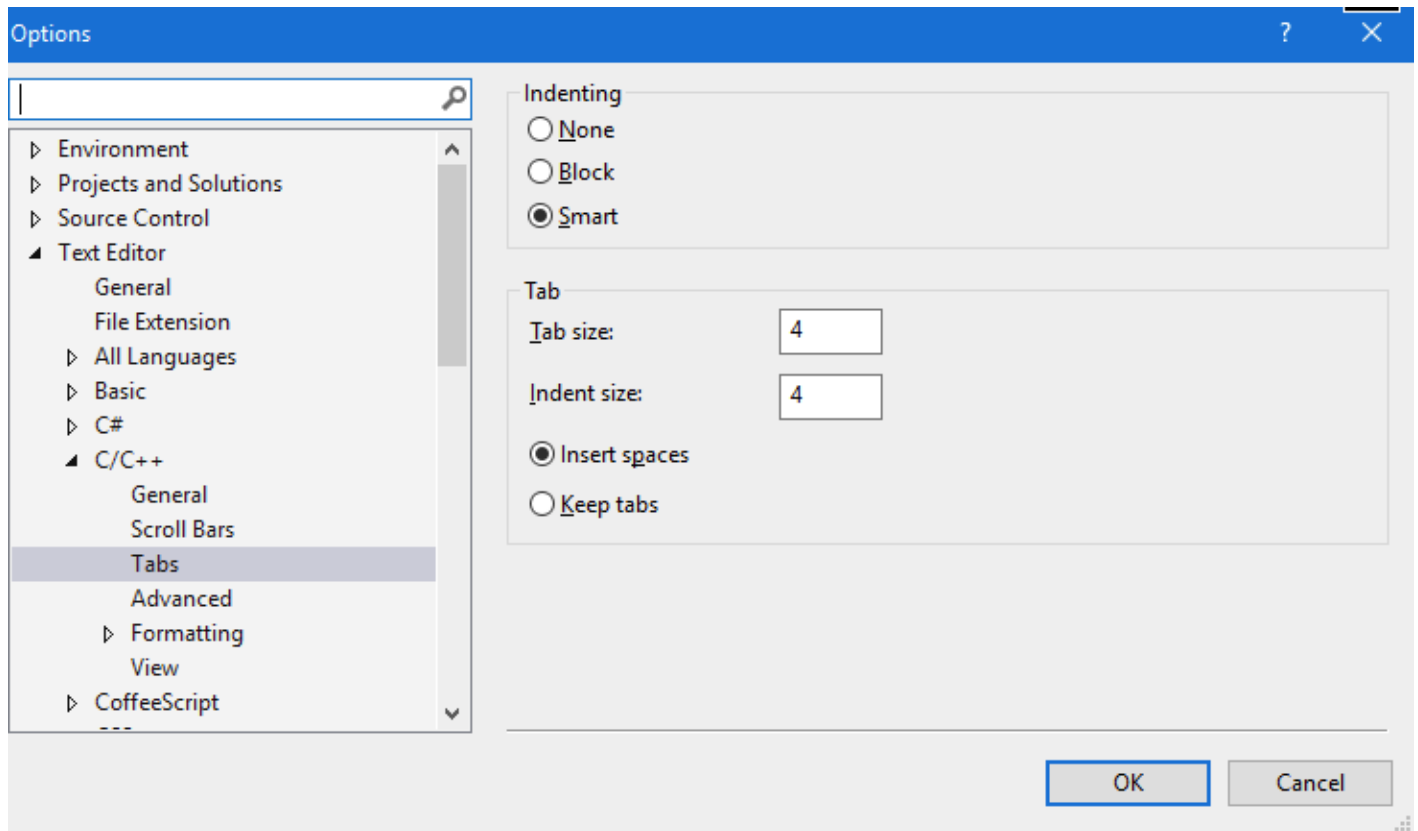


- Select the project 'opencpn' as the 'Startup Project'

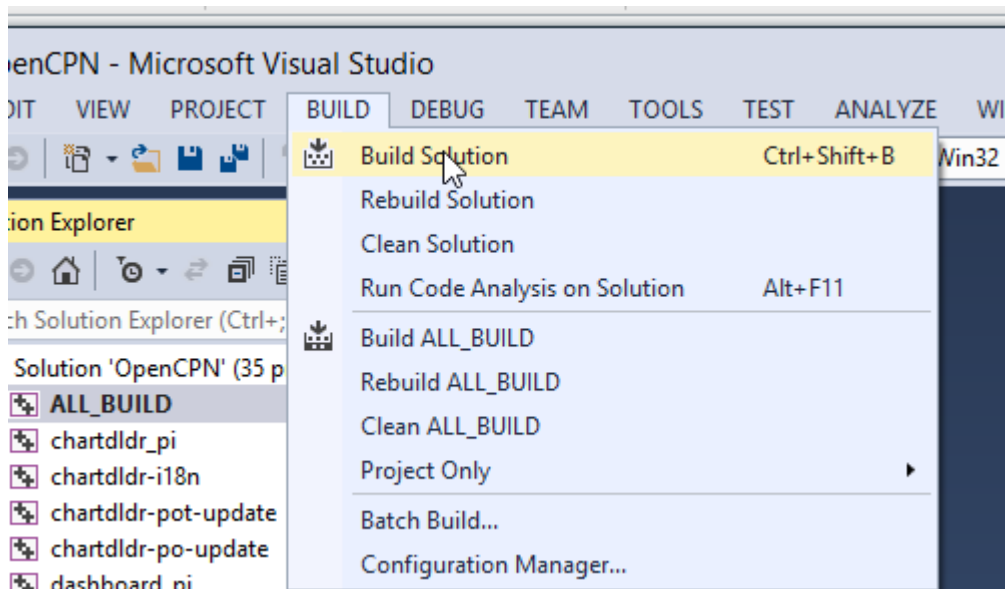


- To avoid problems later please check the following options are set correctly
- Select 'Tools', 'Options'
- Check the settings are the same as shown in these two screenshots

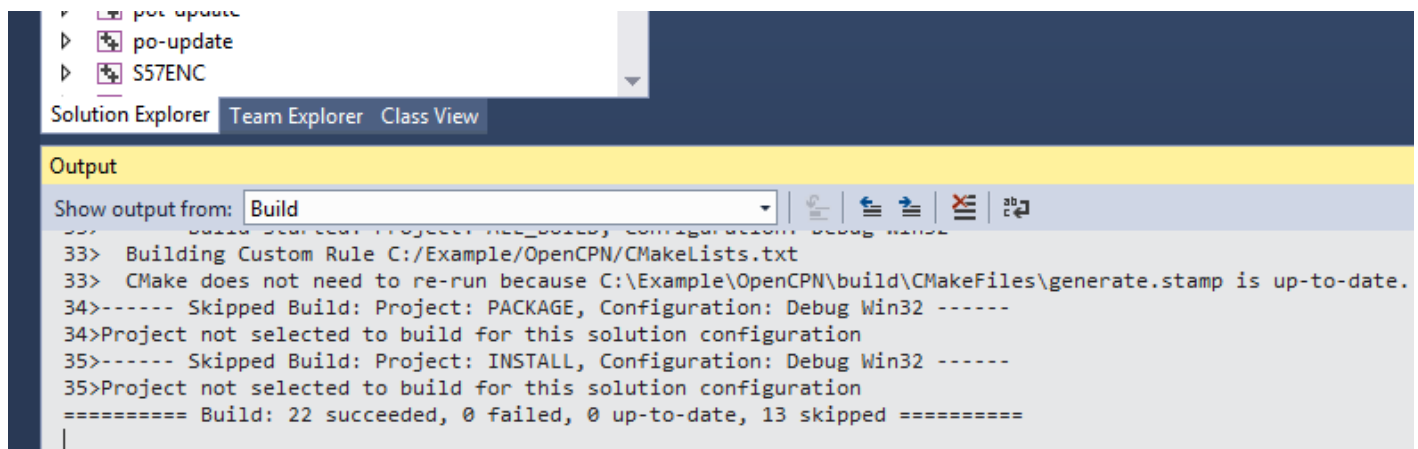




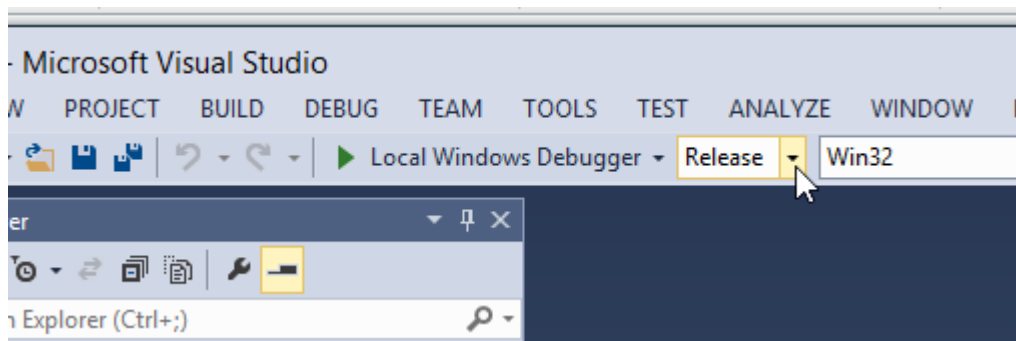
- From the top of the window choose 'Build', 'Solution'.
- You will be making a 'Debug' version of the program.



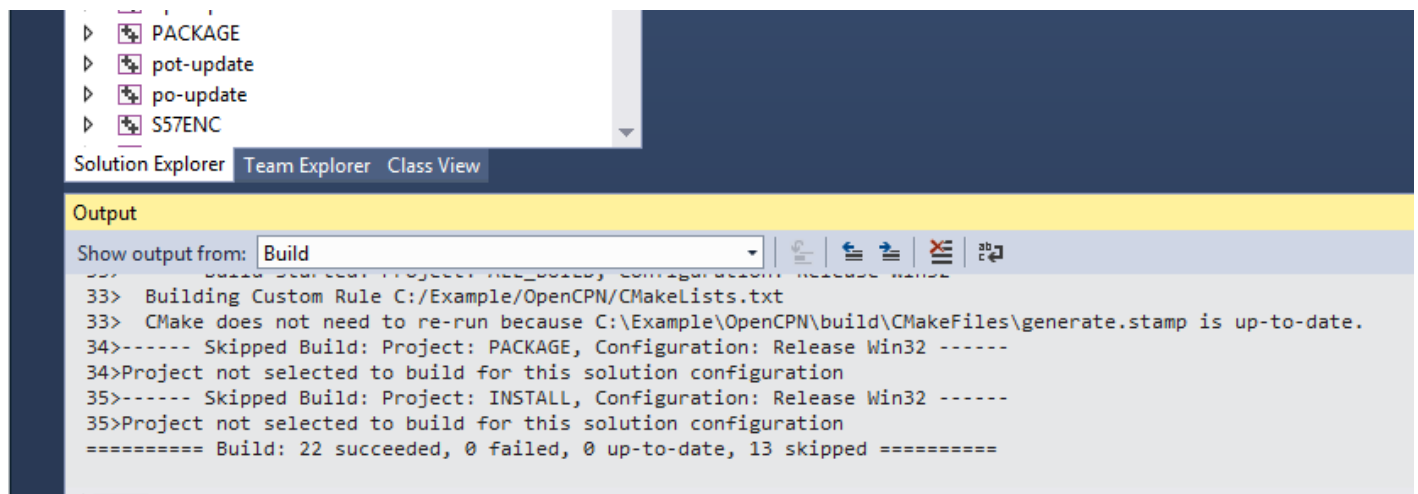
- The build will take some time but the result should be like this:



- All is going well. The release version will now be made.
- Change the 'Dropdown' from 'Debug' to 'Release'.



- From the top of the window choose 'Build', 'Solution' again.

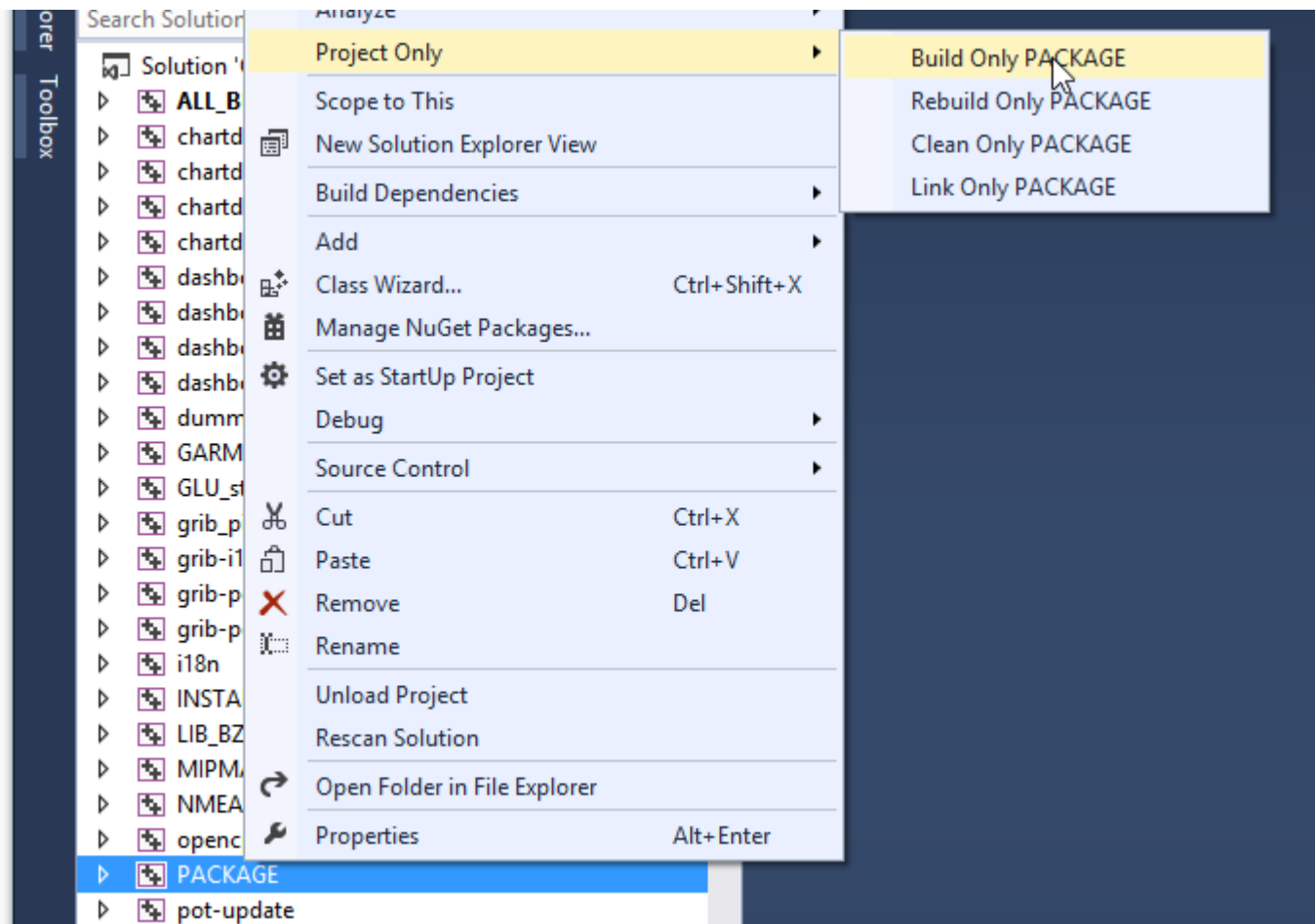


- Two new folders have appeared in 'C:\Example\OpenCPN\build', called 'Release' and 'Debug'.

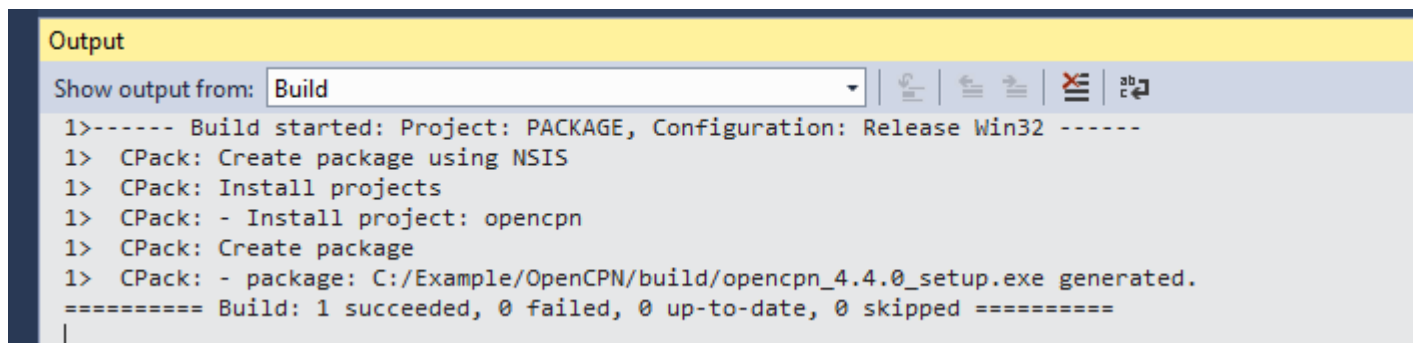
Make a package to install OpenCPN

This assumes that you have installed 'NSIS' (The guide was [here](#)).

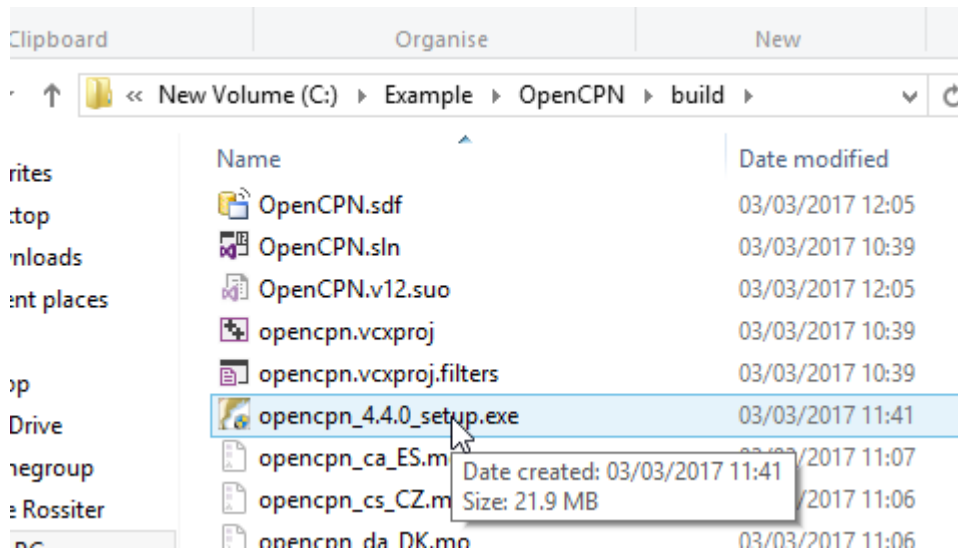
- In Visual Studio Solution Explorer you will see a project called 'PACKAGE'.
- Right-click 'PACKAGE'. Choose 'Project Only', 'Build Only PACKAGE'



- Run this option.

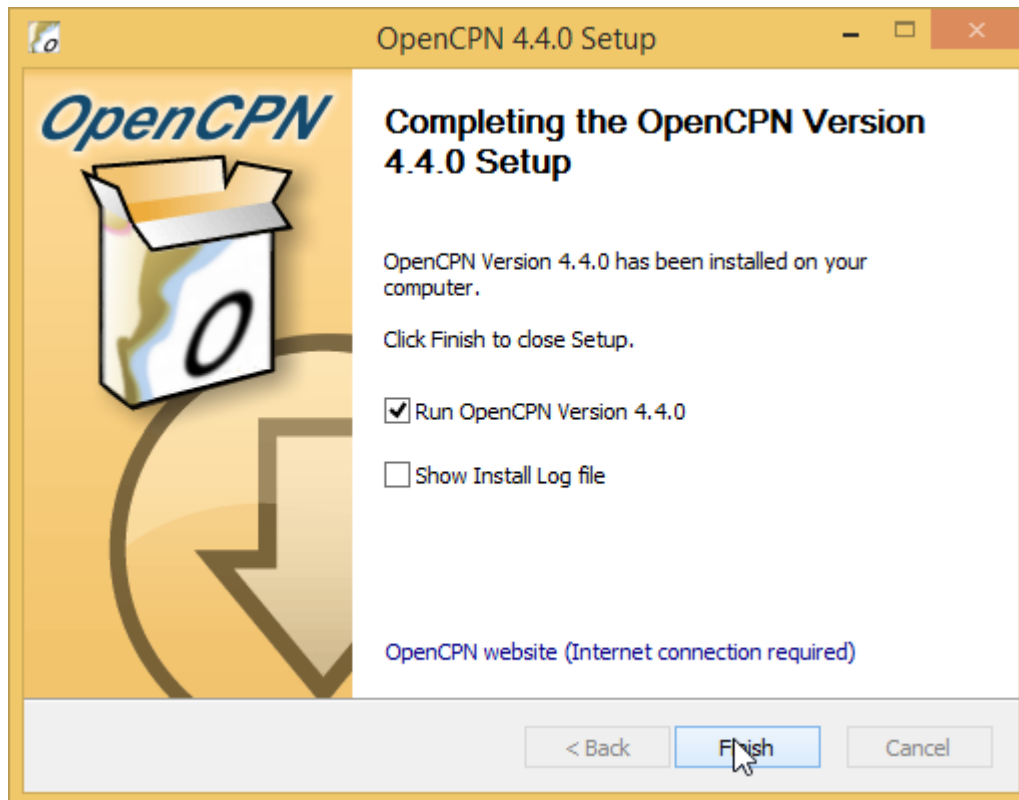


- This will produce an installer 'setup.exe' in the folder 'C:\Example\OpenCPN\build\'

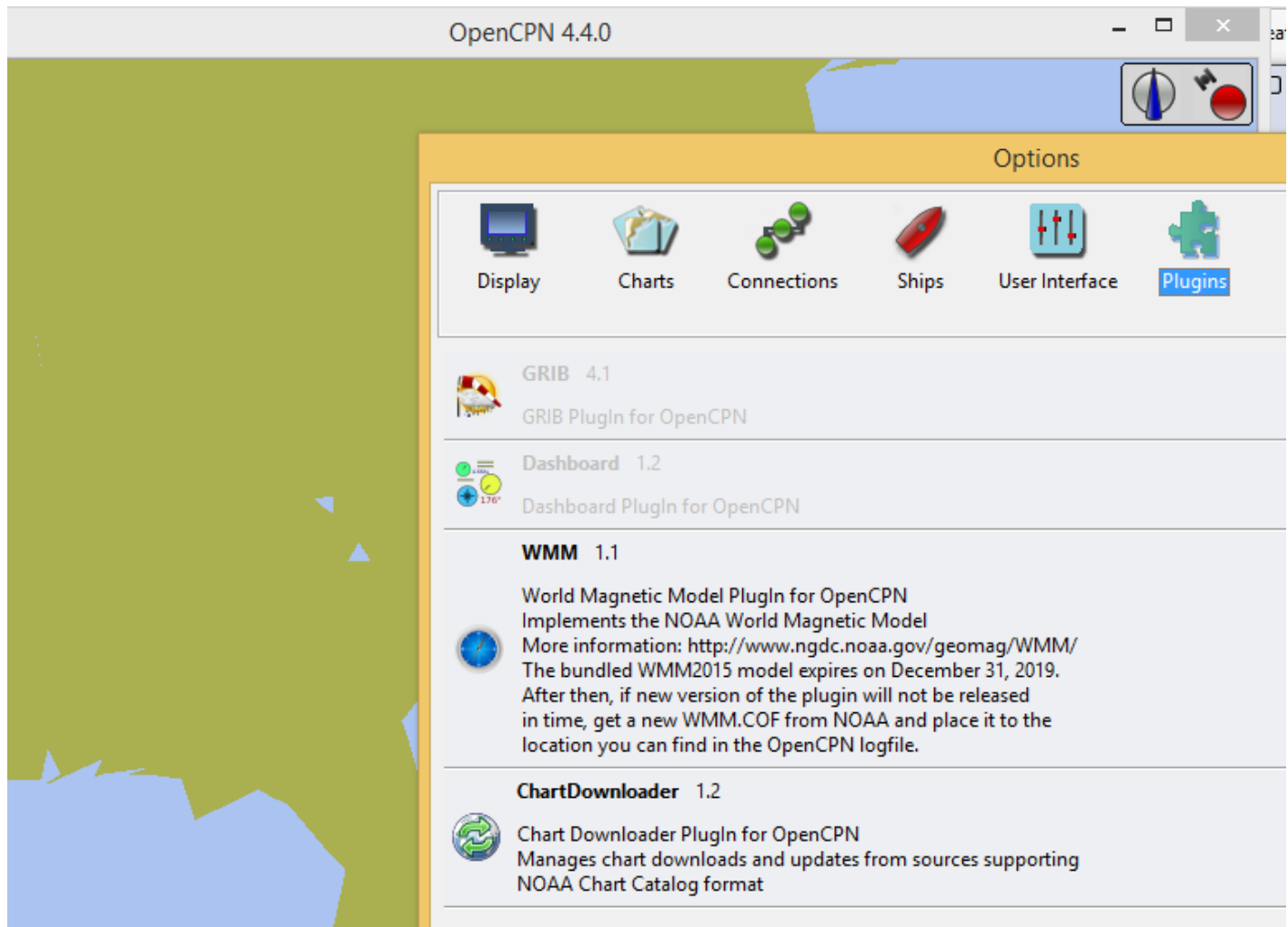


Install OpenCPN

- Double click the setup .exe
- Accept the default settings
- The installation will complete with this page



- Run the finished program



OpenCPN (Version 4.4.0) and the four plugins included in the source files have been built successfully.

From:
<https://opencpn.org/wiki/dokuwiki/> - OpenCPN Manuals

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:developer_guide:compiling_windows:compiling_windows_-_steps_-_example

Last update: 2020/03/07 15:05

